CR 172075

INITIAL DEVELOPMENT OF PASSIVE DOSIMETRY FOR STS APPLICATIONS

by

E. V. Benton, A. L. Frank and R. P. Henke

USF-TR No.71

30 November 1985

FINAL REPORT

Prepared under Contract Number NAS9-15152

Physics Research Laboratory
University of San Francisco
San Francisco, California 94117

for

Lyndon B. Johnson Space Center
National Aeronautics and Space Administration

INITIAL DEVELOPMENT OF PASSIVE DOSIMETRY FOR STS APPLICATIONS

ABSTRACT

This program in dosimetry research and applied space dosimetry has been in effect since December, 1980. During this period three main tasks have been pursued; dosimetry aboard the space shuttles on the Earth-orbiting missions, heavy-ion accelerator beam quality studies and dosimetry support of NASA ground-based radiobiological experiments, and the general development of passive dosimetry techniques with specific goals in their application to space radiation measurements.

On the space shuttle missions both personal and area dosimeters have been deployed. The personal dosimeters measured the radiation doses and ionizing particle quality for the individual astronauts while the larger area dosimeter allowed a more comprehensive sampling of the radiation fields to be made. Since the beginning of the shuttle flights both the personal dosimeters (Crew Passive Dosimeters - CPD's) and the area dosimeter (Area Passive Dosimeter - APD) have undergone changes to reflect the current needs and findings from the previous measurements. The detectors used are all passive and integrating in type. These include plastic nuclear track detectors (PNTD's) for high-in-Z-and-energy (HZE) particles, thermoluminescent detectors (TLD's) for total ionizing particle doses, nuclear emulsions for proton fluences and neutron detectors of the particle radiator/track detector variety to sample different portions of the neutron spectrum. This report reviews and summarizes this effort.

I. INTRODUCTION

The radiation encountered aboard spacecraft in Earth orbit is complex, both in particle type and energy spectra. This reflects the diverse origins of the radiation. There are primary galactic and solar components, either directly incoming or trapped within the radiation belts. These produce secondaries such as recoil nuclei, nuclear reaction products and Bremsstrahlung through interaction with the materials of the spacecraft and its cargo. The fluxes and energy spectra are dependent on altitude and inclination of the orbit, on solar conditions, and amount, type and placement of shielding materials on the spacecraft. For the orbits of the space shuttle flights, the most important component of the primary radiation is from the energetic trapped protons. The higher Z particles yield a smaller but significant component of dose equivalent, depending heavily on the orbit inclination. At higher orbits, such as the geosynchronous orbit required for power stations, trapped electrons become important.

The space environment may prove to be the single most important constraint upon long-term manned space activities. The highly penetrating nature of some components of the space radiation field makes it impractical to provide enough shielding to the crew to completely eliminate the hazard. An indirect hazard also occurs from the effects of radiation on materials and electronics, in addition to the soft errors produced in computers. Biomedical experiments performed in space may need to take possible radiation effects into account. To date, only very limited experimental data exist on the radiation levels and the variation of these levels inside orbiting

spacecraft /1-5/. Although computer codes have been developed for calculating the environment inside the orbiting spacecraft in specific orbits, a number of uncertainties exist including the uncertainties in the proton models (about a factor of 2), in the electron belt models (about a factor of 5), in fragmentation cross-sections of heavy ions, etc. /6, 7/. Moreover, the shielding at any one location within the spacecraft is only approximately known and may vary as the crew and equipment are moved about, consumables used up, and the orientation of the spacecraft changes. It is important to compile an extensive record of radiation measurements aboard the space shuttles, taking in different orbits, but also encompassing repeated measurements in similar orbits.

The accelerator heavy ion beam quality studies and dosimetry support of NASA ground-based radiobiological experiments are performed with PNTD's. Dosimetry is an essential element in radiobiological experiments and the physical aspects of the radiation field must be known because they constitute the independent variable upon which the biological consequences of the radiation are expected to depend. The general objectives are to study the characteristics of accelerator particle beams and to relate these findings to the NASA ground-based radiobiological experiments. More specifically, we (1) measure the linear energy transfer (LET) spectra of typical Bevalac particle beams used in NASA radiobiological experiments and compare the results with active Si-Ge detector measurements being done by University of California-Lawrence Berkeley Laboratory (UCLBL), (2) support the dosimetry of specific NASA radiobiological experiments including the UCLBL "Skyhook" experiment, the NASA-Ames "R.C." experiments and the calibration and testing of the NASA-JSC sponsored "HZE Active Dosimeter," (3) develop and further refine methods to study the LET spectral characteristics

of HZE particle beams as they traverse tissue-equivalent materials, and (4) continue our work on beam fragmentation measurements with the objective of determining specific cross sections for those reactions induced in the elements which are of major importance in tissue studies.

Various sub-projects are important to all the above. These include the development of improved read-out techniques through automation, increased accuracy of feature dimension measurements and better background discrimination; also improvement in the CR-39 PNTD detector material through a determination of the optimum constituents for track detection and the standardization of manufacture. The latter project is carried out in conjunction with commercial fabricators. A prime objective is the development of a CR-39 in which the calibration, in terms of the track parameters as a function of etching procedure, method of read-out and particle LET, remains constant from batch-to-batch.

The general development of passive dosimetry techniques for application to space measurements overlaps extensively with the previous project discussed, since a superior CR-39 plastic, improved read-out, etc., will benefit both space and ground-based measurements. Other areas more specific to space measurements are the continued evaluation and re-design of the CPD's and APD's, improvement of techniques to measure the high-Z, low-energy (HZLE) component of radiation and participation in a coordinated dosimetry program with research groups at other institutions. The objective of the last-mentioned project is to synthesize the experimental and theoretical evaluations of the space radiation environment to achieve improved modelling and prediction.

II. SPACE SHUTTLE DOSIMETRY (OFT)

A. Early STS Flights (STS-1 through STS-4)

On the initial STS flights the CPD and APD dosimeter packets were designed to provide the greatest possible versatility in measuring the cosmic radiation with passive detectors. In the CPD's the TLD modules contained arrays of TLD-700 chips unshielded and shielded by layers of several metals. There were also TLD-600 chips unshielded and shielded by Gd thermal neutron absorbers. Thin PNTD's were placed adjacent to the TLD-600 chips to register alpha particles created in the $^{6}Li(n, ^{4}He)^{3}H$ reaction, and therefore served as thermal and resonance neutron detectors. The different metal shields around the TLD-700 chips have different efficiencies for absorbing the impinging radiation; in particular the soft photons. The possibility of discriminating between portions of the radiation environment on the basis of attenuation was therefore tested with this TLD array.

The CPD HZE stacks contained 14 layers of PNTD's of three different types. The three types, CR-39, cellulose nitrate (CN) and Lexan polycarbonate (PC), have different sensitivities for detecting HZE particles. Together they constitute a stack capable of measuring fluences of cosmic ray HZE particles over a wide range of energies, atomic numbers and intensities. The LET spectra of the HZE particles can be measured separately on each type for the different low LET cutoffs corresponding to the PNTD sensitivities.

The physical configurations of the TLD module and the HZE stacks are shown in the schematics in Figures 1 and 2. The components in Fig. 1 are labelled as follows, with the thicknesses of the materials also listed:

Al	aluminum (0.159 cm)
CN	cellulose nitrate (0.01 cm)
Cu	copper (0.159 cm)
Gd	gadolimium (0.0025 cm)
P	paper (0.02 cm)
Pb	lead (0.159 cm)
PC	polycarbonate (0.159 cm and 0.025 cm)
2	TLD-200 calcium fluoride (0.089 cm)
6	TLD-600 lithium-6 fluoride (0.089 cm)
7	TLD-700 lithium-7 fluoride (0.089)

The plastic sheets of the LET stack are arranged as shown in Fig. 2. The sheets are numbered as shown, where MM is the two-digit mission number, SS is the two-digit LET stack number, and LL is the two-digit layer number. The full sequence of layers in the stack is: CR-39, PC, CN, PC, CN, PC, PC, PC, PC, CN, PC, CN, PC, CR-39. The dimensions of the stacks were 6.5 cm x 4.5 cm x 0.45 cm.

The TLD module and HZE stack were enclosed together in a close-fitting polyethylene bag and separated by a thin strip of foam plastic for flexibility. The final overall dimensions of the CPD's were approximately 11 cm x 6 cm x 0.7 cm. During the shuttle flights the CPD's were worn in the left breast pockets of the astronauts.

The APD's were fire-retardant Lexan boxes, 10 cm x 10 cm x 5 cm in dimensions, which contained nine sub-components (Fig. 3). Five of the sub-components (E to I) consisted of five stacks of PNTD's oriented with equal stack volumes in the three orthogonal directions. The plastic stacks contained alternating layers of three different detector types. A single repeat unit contained, in order, four layers of the plastic types, GE Lexan polycarbonate, American Acrylics CR-39, Lexan, and Kodak Pathé cellulose nitrate (CN), with respective nominal layer

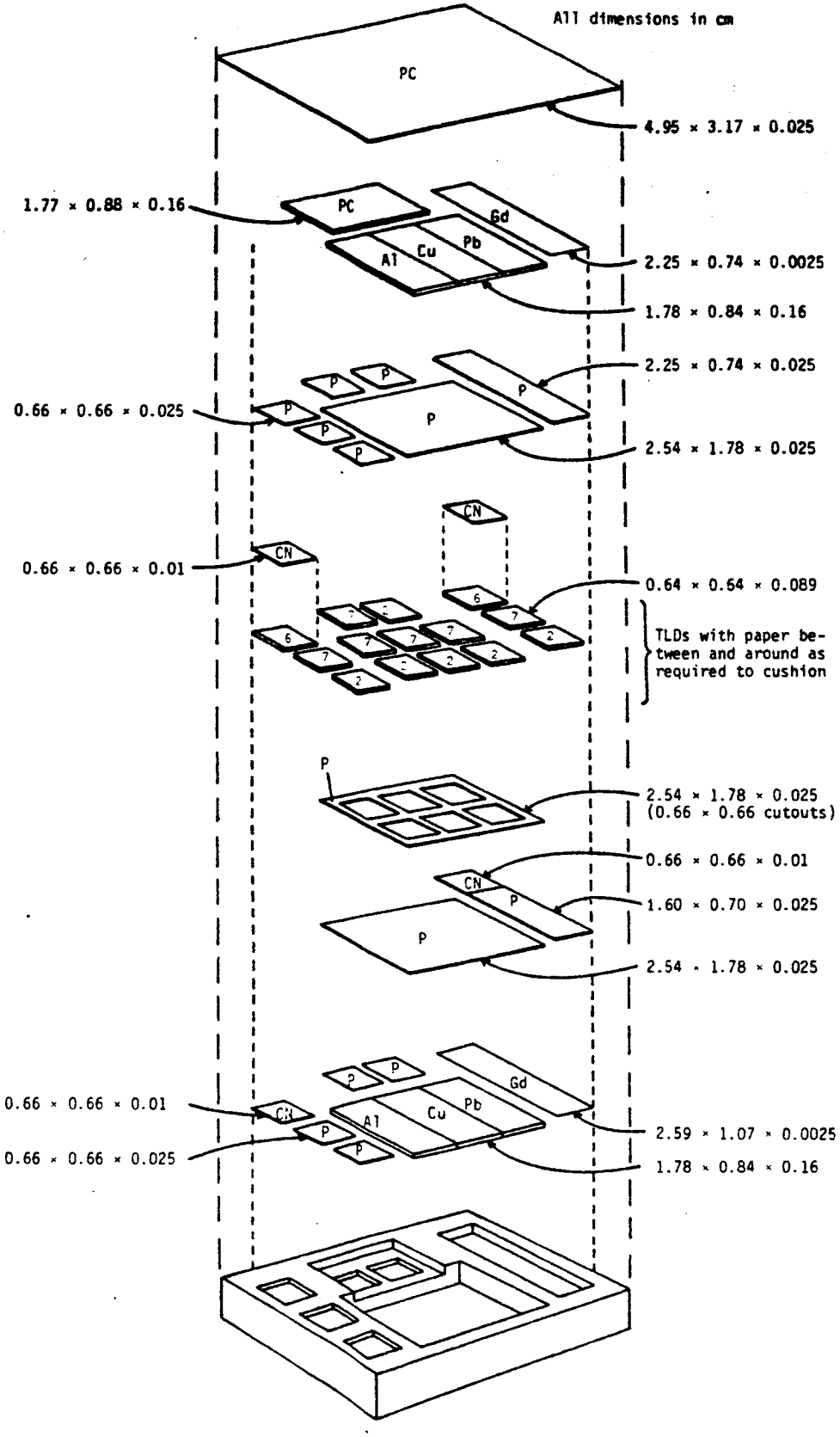


Fig. 1 TLD module

QA

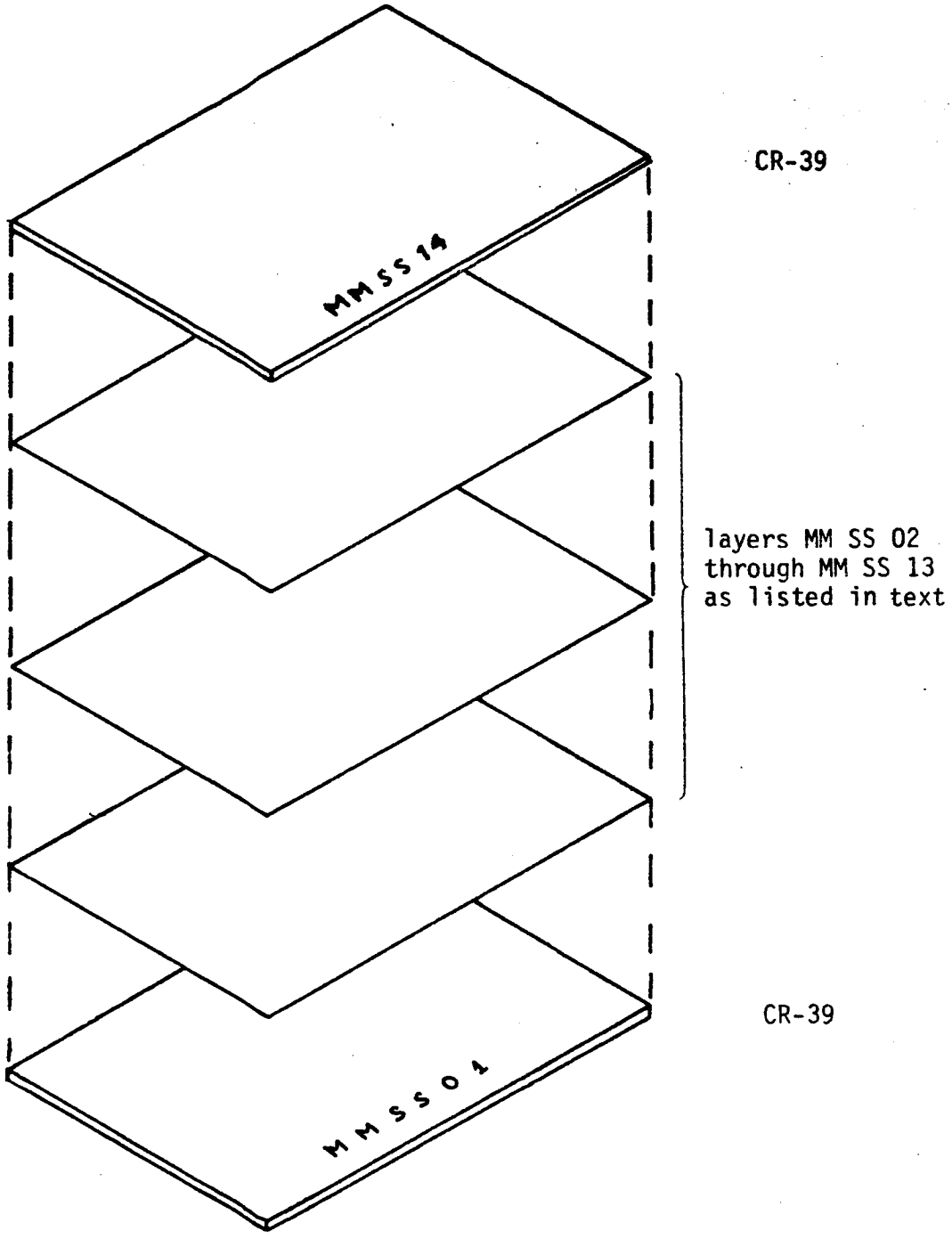
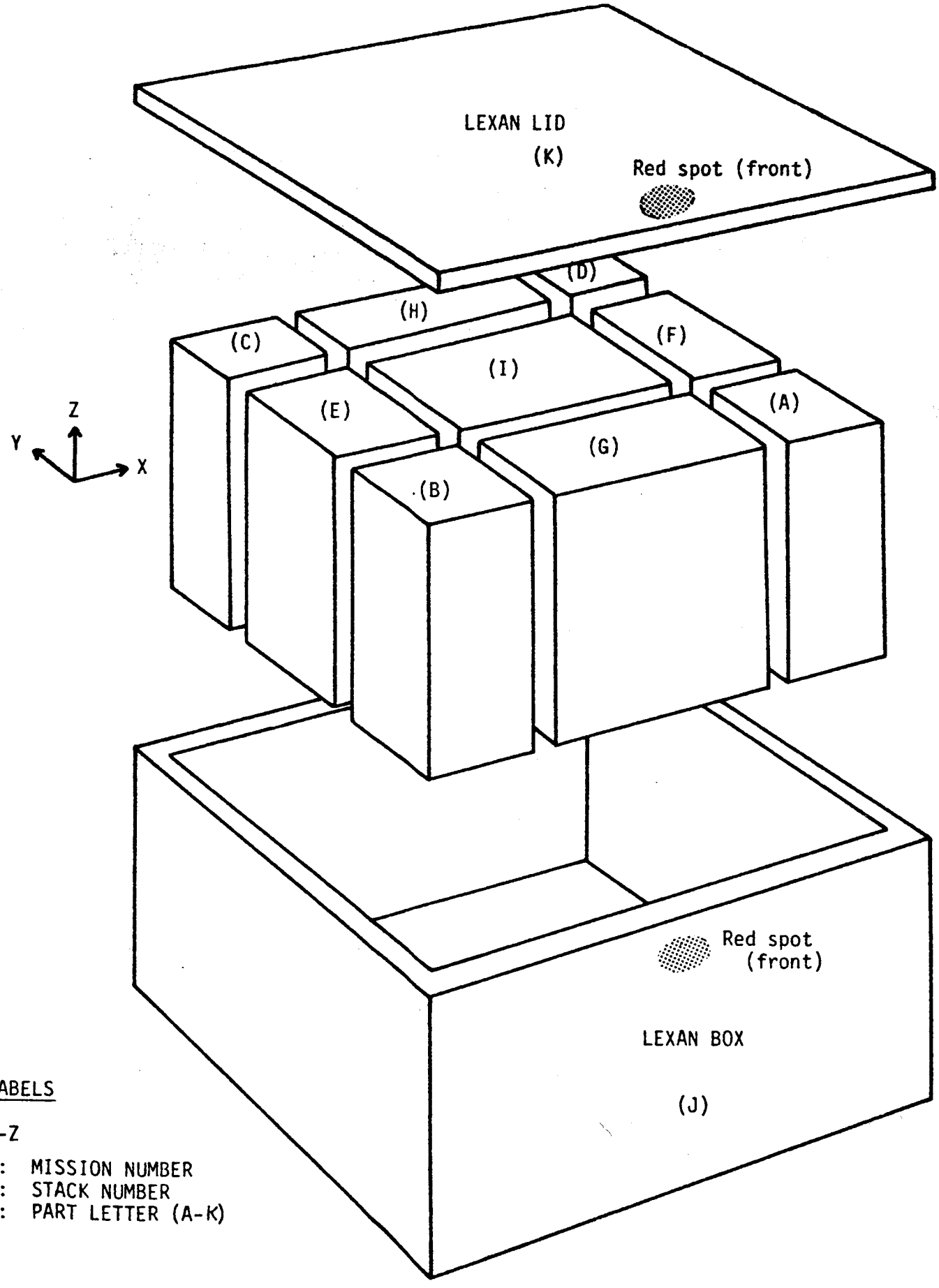


Figure 2. LET stack.



MODULE LABELS

MM-SS-Z

MM: MISSION NUMBER
SS: STACK NUMBER
Z: PART LETTER (A-K)

Fig. 3. Area Passive Dosimeter package

thicknesses of 190 μm, 1000 μm, 190 μm, and 100 μm. This sequence was repeated a sufficient number of times to comprise the entire thickness of each of the stacks. The arrangement of the plastic stacks along the three orthogonal directions is the preferred configuration because of the somewhat anisotropic response of the detectors (they respond to lower LET particles at normal incidence than at grazing incidence).

The corner stacks (A, B, C and D in Fig.3) were designed to hold both TLD's, for measuring the total absorbed dose, and particle radiation detectors, for measuring neutron fluences. The TLD's employed were TLD-200 (calcium fluoride) and TLD-700 (lithium-7 fluoride). The particle radiators were of two types. Foils of lithium-6 fluoride were used to produce ^{4}He particles by the $(n,^{4}He)$ reaction. CR-39 plastic was sandwiched against the radiator foils to record the ^{4}He particle tracks. Fission foils of ^{209}Bi, ^{232}Th and ^{238}U were used to produce fision fragments by the (n, f) reaction. Moscovite mica was used to record the tracks of the emitted fission fragments.

It was necessary to separate the radioactive fission foils from the TLD's in the APD in order to reduce the build-up of background gamma radiation doses in the TLD's as much as possible. The corner stacks were therefore built in two different configurations. Stacks A and C were designed to hold the fission foil detectors plus those using ^{6}Li fluoride foils; stacks B and D were designed to hold the TLD's. Stacks A and C were composed of polycarbonate plates with alternating plates having milled holes of the proper sizes to contain the radiator detectors. Each stack contained two ^{6}Li fluoride foils, 4.5 mg/cm^2 in thickness and 1.6 cm in diameter, on a filter paper backing. One of these foils was shielded by 0.0025 cm-thick Gadolinium foils. The following unbacked foils were also

placed in each of stacks A and C: one ^{209}Bi (3.8 cm x 1.3 cm x 0.12 cm), three ^{238}U foils (each 1.3 cm in diameter and 0.025 cm thick), and two ^{232}Th foils (each 1.3 cm in diameter). The three preceding foil types had mica detectors placed against both foil surfaces. The ^{232}Th foils, along with their mica detectors, were additionally individually sealed in 1.6 cm diameter polyethelene pouches. This prevented the escape of any ^{220}Rn produced in the foil to the surrounding detectors.

Stacks B and D contained the TLD-200 and TLD-700 chips (each 0.65 cm x 0.65 cm x 0.09 cm). Polycarbonate blocks separated the three milled polycarbonate plates which housed three TLD-200 and three TLD-700 chips each. The three sets of TLD's were placed near the top, bottom and center of each stack.

B. Detector Calibrations

The TLD's were routinely calibrated with a standard ^{137}Cs gamma-ray source. For each flight the background, calibration and flight chips, from single manufactured batches of TLD's and with similar sensitivities, were annealed together. The annealing method used was to place the TLD's in an oven at 400°C for 1.5 hrs., then allow them to return to room temperature gradually, over several hours. Near the mid-point of the space shuttle missions, the calibrations were made with the TLD's placed behind 1.75 mm of plastic. The Cs source used was cross-calibrated against several other calibration sources whose activity was traceable to NBS standards.

In addition to the above, a series of calibrations to X-ray photons of different mean energies were made with TLD's loaded into the CPD's. These irradiations were made to test the effectiveness of the different absorbers and TLD types for the purpose of discriminating among incident radiations of different

absorption coefficients. The exposures were made at a standard exposure facility /8/ with the CPD's mounted on phantoms and with normally incident X-rays. The results are shown in Table 1 where the TLD responses to the X-rays are compared to those from ^{137}Cs gamma rays. The over-responses of the TLD's at these photon energies, for the unshielded and lightly shielded cases, and the attenuations of the different shields are clearly seen.

The calibration for the PNTD's in the HZE stacks were performed using accelerated heavy ions at the Berkeley Bevalac. The dependence of the etched track parameters on the particle LET's was determined, which allows reconstruction of LET spectra from experimental track measurements. This also allows the determination of the effective low-LET cutoff employed in the scanning of the experimental samples, which is critical for an accurate measure of heavy particle doses.

The sensitivity of a PNTD is expressed in terms of V_T/V_G versus LET_W, where V_T is the etch rate along the track in the PNTD, V_G is the bulk etch rate of the material and LET_W denotes that part of the energy loss rate in the material due to electrons of energy less than W eV (where $LET_\infty = dE/dx$). Physically, this means that only the lower energy recoil electrons along the trajectory of the HZE particle deposit their energy into the core region of the track and contribute to an increase in the etch rate of this damaged material. In CR-39 the W-value is about 200 eV.

The neutron detectors were calibrated by experimental and calculational techniques. In the CPD's, where the TLD-600 (^6LiF) chips were used as alpha particle radiators to CN PNTD's by the reaction $^6Li(n,T)^4He$, the detectors were basically 2π detectors for thermal neutrons. The 2π efficiency of these detec-

TABLE 1

Response of CPD TLDs to X-rays*

TLD type	Shield	Mean X-ray Energies 19.7 keV	53.5 keV	91.1 keV
TLD-200	bare	5.13 ± 0.01	11.8 ± 0.50	7.07 ± 0.25
TLD-700	bare	1.49 ± 0.04	1.59 ± 0.06	1.43 ± 0.05
TLD-600	bare	1.38 ± 0.02	1.55 ± 0.04	1.38 ± 0.06
TLD-200	Gd	2.57 ± 0.34	10.2 ± 0.4	6.58 ± 0.19
TLD-700	Gd	0.49 ± 0.01	1.43 ± 0.01	1.40 ± 0.04
TLD-600	Gd	0.46 ± 0.03	1.38 ± 0.01	1.43 ± 0.01
TLD-200	plastic	2.62 ± 0.04	10.2 ± 0.4	6.18 ± 0.69
TLD-700	plastic	0.79 ± 0.01	1.57 ± 0.01	1.42 ± 0.04
TLD-200	Al	1.14 ± 0.02	9.30 ± 0.17	6.27 ± 0.46
TLD-700	Al	0.22 ± 0.01	1.25 ± 0.01	1.31 ± 0.03
TLD-200	Cu	0.018 ± 0.003	1.11 ± 0.08	1.79 ± 0.10
TLD-700	Cu	0.013 ± 0.001	0.20 ± 0.01	0.61 ± 0.02
TLD-200	Pb	0.032 ± 0.002	0.49 ± 0.05	0.46 ± 0.01
TLD-700	Pb	0.020 ± 0.001	0.073 ± 0.005	0.12 ± 0.01

*Relative to ^{137}Cs irradiated under the standard conditions referred to in text.

tors has been found to be /9/:

$$E_{2\pi} = 5.6 \times 10^{-3} \text{ tracks/thermal neutron} \qquad (1)$$

The efficiency for a 4π fluence is then half this, or 2.8×10^{-3} tracks/thermal neutron.

In the APD's, thin films of ^{6}LiF (4.5 mg/cm^2) were used as alpha particle radiators to CR-39 PNTD's. Since thermal neutrons could penetrate through the back side (through the ^{6}LiF) in these detectors, a higher efficiency of 4.9×10^{-3} tracks/thermal neutron resulted. The separation between thermal (<0.2 eV) and resonance (0.2 eV–1 MeV) neutrons was made with the gadolinium absorber. The fractional penetration of isotropic neutrons through the Gd foil, as a funtion of energy, is shown in Fig. 4. The sensitivity for resonance neutrons, where a 1/En spectrum is assumed, has been calculated to be 2.56×10^{-4} tracks/neutron.

The method of calibration for the fission foil/mica detectors has been previously described in reports on space dosimetry on the Cosmos 936 and 1129 flights /10, 11/. The track densities contain both proton-induced and neutron-induced components. The fraction due to neutrons and the neutron sensitivity are functions of the proton and neutron fluences and spectra present on the spacecraft. These quantities were not measured during the STS flights, but a method has previously been developed which requires approximations to be made for the spectral shapes and relative fluences based on past space flights and calculations of neutron production in the atmosphere. Due to the nature of the assumptions made in the high energy neutron (>1 MeV) determination, the associated errors can only be estimated and could be rather large.

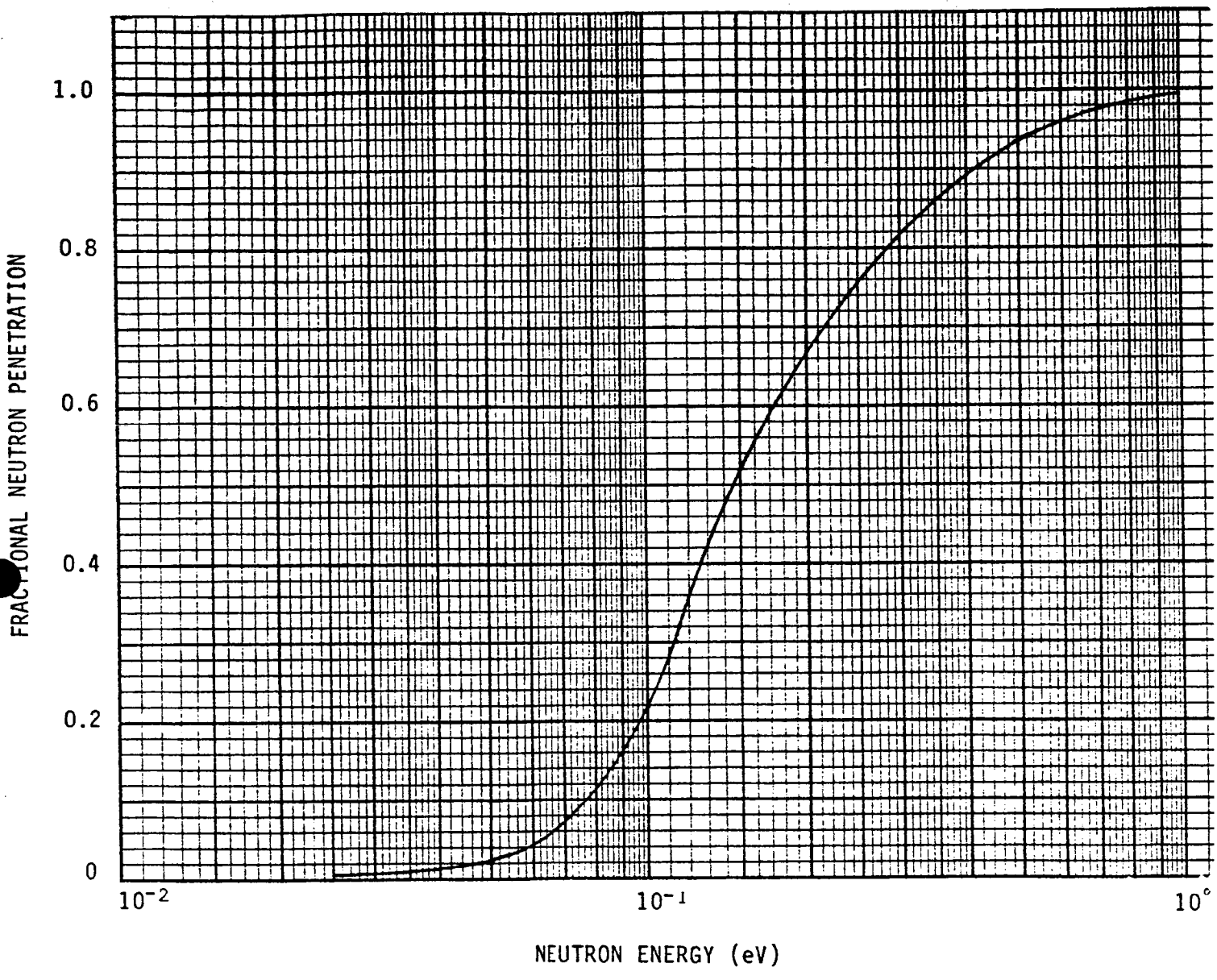


Fig. 4 The penetration of isotropically incident neutrons through gadolinium foil of 0.0025 cm thickness as a function of neutron energy.

The fluence-to-dose conversion factors for the thermal and resonance neutrons are from the NCRP* (1971) values, where the resonance value was weighted tor a 1/E spectrum. For the high energy neutrons, a Quality Factor of 10 was assumed.

C. Dosimeter Design Changes after OFT

Information acquired during measurements on the first four STS flights (OFT 1–4) was used to re-evaluate the performance of the CPD's and APD's. Design changes were made to improve the quality of the dosimeter measurements by eliminating components of marginal value and upgrading the more useful parts. The original designs were for maximum flexibility in covering wide possibilities in radiation doses, fluences and spectra. The re-design was aimed at improving the measurements for the particular radiation characteristics encountered on the STS missions. Also some fundamental improvements were incorporated.

1. CPD design changes.

Beginning with STS-6, the TLD module of the CPD's was simplified. The series of absorbers and the TLD-200 chips were eliminated while the number of unshielded TLD-700 chips was increased to five. The statistical reliability of doses measured with the TLD-700 chips was increased by this change, while the shielded TLD data had not proven to be of value in evaluating radiation quality.

An example of the shielded TLD data, from STS-3, is given in Table 2 for eight flight CPD's. This is typical of all measurements made with this detector configuration. It is seen that the shields serve to increase the TLD doses, in correspondence with mass density and atomic number of the shielding materials.

*National Council on Radiation Protection and Measurement

TABLE 2. FLIGHT DOSES FROM STS-3 CPD's

TLD #	TLD Type	Shield	CPD Flight Doses (mrem)							
			301	302	305	306	307	308	309	310
1	200		41.5	45.0	48.7	46.1	40.7	46.1	50.4	45.2
2	700		47.1	45.9	49.0	46.2	43.4	50.2	44.4	46.0
3	600		53.1	46.6	59.2	50.2	48.1	61.3	51.8	52.7
4	200	Gd	42.7	44.4	46.0	47.3	45.1	46.2	45.1	46.1
5	700	Gd	46.2	49.0	48.3	46.5	48.7	48.3	48.1	53.2
6	600	Gd	51.3	44.4	47.4	46.4	43.6	46.9	51.1	49.6
7	200	Lexan	46.4	50.3	46.5	46.8	46.7	45.2	44.5	46.4
8	700	Lexan	44.7	45.7	46.5	45.5	43.4	49.3	44.3	48.5
9	200	Al	46.5	44.6	47.6	46.8	43.9	47.9	43.8	47.4
10	700	Al	49.0	44.7	47.3	46.7	45.7	49.9	45.1	45.5
11	200	Cu	51.2	48.5	46.3	54.4	49.7	51.0	48.1	50.9
12	700	Cu	50.6	51.2	54.9	50.4	50.2	55.9	47.6	51.8
13	200	Pb	57.5	59.0	57.6	54.0	56.8	60.0	52.2	53.3
14	700	Pb	55.3	59.8	66.8	57.3	59.1	60.4	56.7	57.5

This implies that radiations which would be attenuated by the shields are not an important component of total astronaut dose for the orbits of these missions. The increase seen in the shielding doses must be an effect of high energy proton interaction with the shielding materials, since trapped protons dominate the total doses. A general reduction in energy of the protons in traversing the shields would result in higher LET's and total energy imparted to the TLD's. Also, nuclear interactions by the protons can result in emissions of short range, highly ionizing particles by the shield nuclei.

Beginning with STS-7, the TLD modules in the CPD's were furnished by NASA-Lyndon B. Johnson Space Center personnel.

Beginning with STS-6, the PNTD's in the HZE stacks were reduced in number and given the configuration shown in Fig. 5. It was found that the HZE particle track densities were best measured by the CR-39 detectors. Two Lexan sheets were retained for physical protection of the stacks and a CN sheet was kept for a back-up. The two CR-39 sheets were placed together to take advantage of an improved method of track scanning. The selected HZE events are those consisting of coincidences across a CR-39/CR-39 interface. This gives a much more sound rejection of background tracks produced by radon and radon daughter alpha particle events than the subtraction of a background count on a ground control sample. This method also does not compound the counting statistical errors as a background subtraction does.

Beginning with STS-51C, the CN back-up sheets in the HZE stacks were replaced by a second pair of CR-39 sheets. This provided backups of greater potential value in HZE particle measurements than the CN detectors.

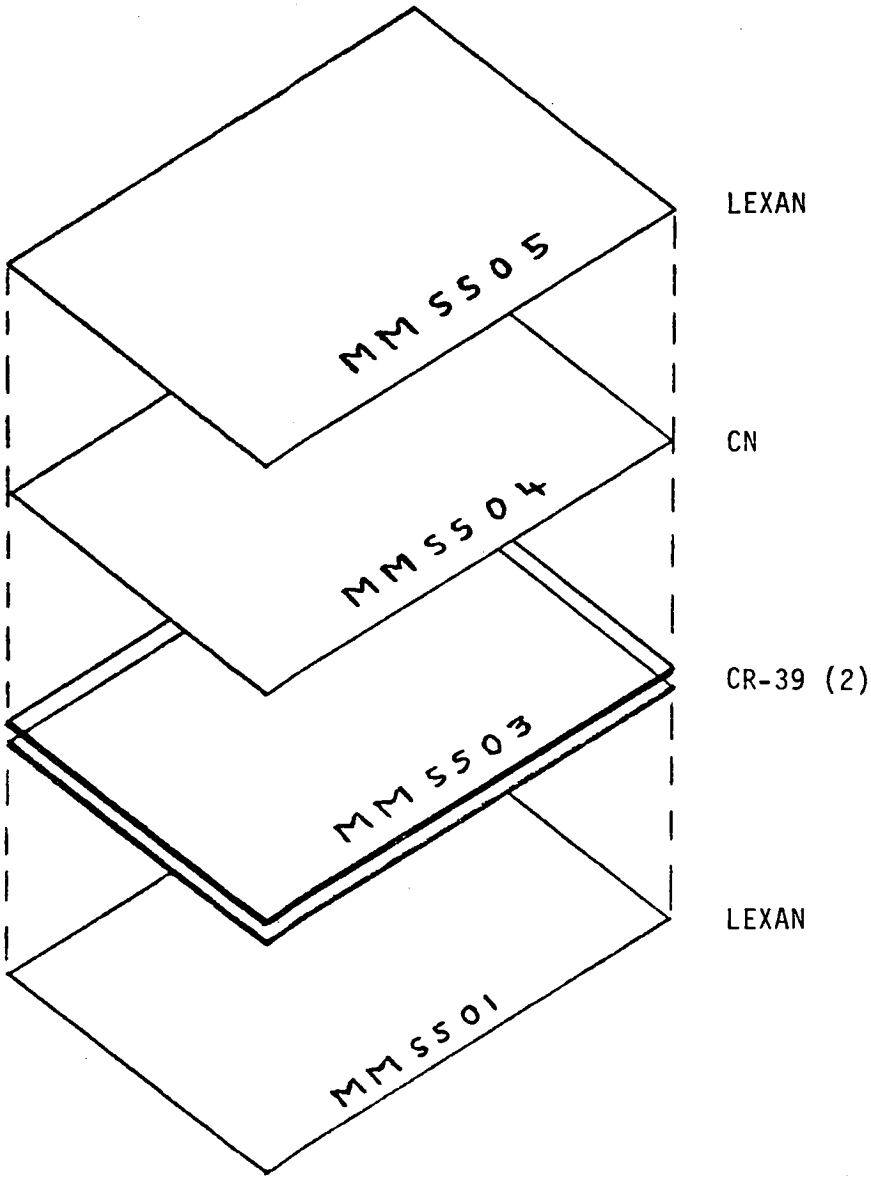


Fig. 5. LET stack

2. APD Design changes.

Beginning with STS-5, the APD configuration was changed to that shown in Fig. 6. Six thin HZE stacks, of the type previously described in the CPD section above (Fig. 5), were arrayed at the six faces of the APD box. This preserved the advantage of having the directionally dependent PNTD's arrayed in the three orthogonal directions while allowing the elimination of the thick plastic stacks. The thick stacks used very large amounts of PNTD materials, only a small amount of which was actually needed. As with the CPD HZE stacks, beginning with STS-51C second pairs of CR-39 PNTD's were added to the stacks as back-up detectors while the CN sheets were removed.

The four corner stacks of the original design were grouped in two; one for TLD's and one for neutron detectors. During the first four STS flights we established that the TLD arrays being used did not provide information on radiation attenuation through the APD's. The lockers where the APD's were stored were shielded sufficiently that the shorter range radiations from space did not penetrate to the APD's. The more penetrating radiations which were measured by the detectors yielded essentially equal doses to all the TLD's.

The ^{209}Bi/mica and ^{238}U/mica detectors were also eliminated from the neutron detector stack. The ^{209}Bi has a very small cross section and the ^{238}U foils emit a significant spontaneous fission background. On the STS flights, where neutron fluences are relatively low, neither of these detectors yields track densities significantly above background.

Beginning with STS-41C the ^{232}Th/mica detectors were eliminated, leaving only the ^{6}LiF/CR-39 thermal and resonance neutron detectors. This was requested by NASA-JSC personnel in order to reduce any possibility of a false exposure from

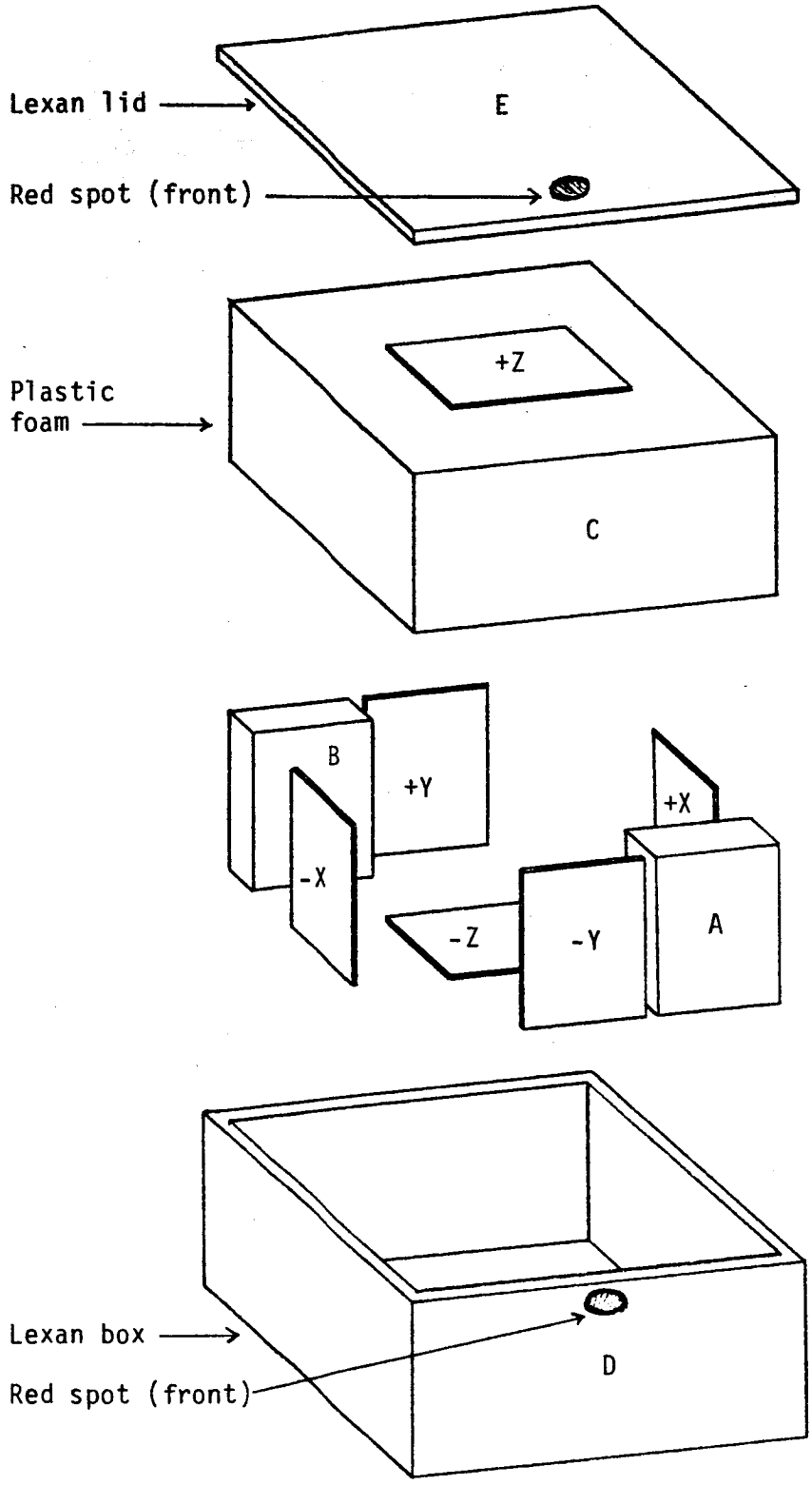


Fig. 6. Schematic diagram of the APD package. The module labels follow the convention MMMDDL where MMM = mission number, DD = dosimeter number and L = part letter (A to D and -X to +Z).

the radioactive foils. Beginning with STS-51C, all neutron detectors were eliminated from the APD's, since the high energy neutrons contribute most of the neutron dose while the thermal and resonance neutron doses are relatively small.

After STS-51C, the APD's were no longer flown on a regular basis; however, a fairly large body of meaurements had been acquired during earlier flights.

D. STS Dosimetry Measurements

The dose measurements made on the space shuttle are summarized in Table 3. The doses were measured in the APD's but the CPD doses, TLD's through STS-6 and HZE stacks through STS-51C, were essentially the same as the APD's.

1. Low LET dose

The low-LET doses given in the table are the TLD-700 doses with the high-LET doses, as measured with HZE stacks, subtracted out. The LET efficiencies of TLD-700 have been measured (Fig. 7) so that a knowledge of the high LET spectrum accumulated during a mission allows the high LET contribution to the TLD dose to be calculated.

2. HZE measurements

The HZE fluences and LET spectra were all obtained from CR-39 PNTD's, since these detectors have superior sensitivity and resolution over the entire LET range of the particles which were detected. As mentioned earlier, for the initial flights single sheets of CR-39 were scanned and non-flight control samples were used to subtract out background events. The presence of radon and its daughter products, which are alpha particle emitters, in the ambient air means that stored PNTD's acquire a background of real tracks. There are also spurious pits which can appear to be tracks. When two adjacent layers of CR-39 from the HZE stacks are re-aligned and the inner surfaces scanned together, one finds a

TABLE 3. SPACE SHUTTLE DOSIMETRY SUMMARY

	Whole-Body Dose Equivalents (mrem)			
	STS-1	STS-2	STS-3	STS-4
LOW-LET*		12.5 ± 1.8	52.5 ± 1.8	44.6 ± 1.1
Rate (/day)		5.2 ± 0.8	6.5 ± 0.2	6.3 ± 0.2
Neutron				
Thermal	< 0.05	< 0.03	0.03	0.04
Resonance	< 0.75	< 0.3	2.0	1.6
High Energy	----	----	7.7	14
Total	< 15	< 6	9.7	15.6
HIGH-LET**	3.6 ± 0.4	1.0 ± 0.4	6.3 ± 1.0	7.7 ± 2.9
Total Mission Dose Equivalent		< 19	68.5	67.9
Mission Parameters				
Storage Locker				
Duration (hrs)	34	57.5	194.5	169.1
Inclination (deg)	38	38	40.3	28.5
Altitude (km)	140	240	280	297

	STS-5	STS-6	STS-7	STS-8
LOW-LET*	27.8 ± 2.5	27.3 ± 0.9	34.8 ± 2.3	34.8 ± 1.3
Rate (/day)	5.6 ± 0.5	5.5 ± 0.2	5.8 ± 0.4	5.8 ± 0.2
Neutron				
Thermal	0.03	0.03	0.02	0.02
Resonance	0.7	1.9	1.4	2.6
High Energy	11	6.5	----	----
Total	11.7	8.4	1.4***	2.6***
HIGH-LET**	14.5 ± 1.6	13.8 ± 1.8	11.7 ± 1.6	19.2 ± 3.5
Total Mission Dose Equivalent	54.0	49.5	47.9***	56.6 3.7***
Mission Parameters				
Storage Locker	MF140	MF28K	MF28K	MA16F
Duration (hrs)	120	120	143	70 75
Inclination (deg)	28.5	28.5	28.5	28.5
Altitude (km)	297	284	297	297 222

*Photons and electrons of any energies. High-LET at lower efficiency.
**HZE particles with LET > 20 keV/μm of water.
***Does not include high-energy neutron dose.

TABLE 3. SPACE SHUTTLE DOSIMETRY SUMMARY (continued)

	Whole-Body Dose Equivalents (mrem)			
	STS-9	STS-41B	STS-41C	STS-41D
LOW-LET*	101.1 ± 3.1	43.6 ± 1.8	403 ± 12	42.0 ± 2.8
Rate (/day)	10.1 ± 0.3	5.5 ± 0.2	57.6 ± 1.7	7.0 ± 0.5
Neutron				
Thermal	0.1	0.02	0.05	0.01
Resonance	2.2	0.5	3.1	1.5
Total***	2.3	0.5	3.2	1.5
HIGH-LET**	76.3 ± 9.2	13.6 ± 1.5	98 ± 3	21.3 ± 1.3
Total Mission Dose Equivalent**	179.7 ± 9.7	57.7 ± 2.3	504 ± 12	64.8 ± 3.1
Mission Parameters				
Storage Locker	MF28E	MF280	MF280	MF280
Duration (hrs)	240	191	168	145
Inclination (deg)	57	28.5	28.5	28.5
Altitude (km)	241	297	519	297

	STS-41G	STS-51A	STS-51C
LOW-LET*	82.4 ± 2.4	94.3 ± 4.9	35.4 ± 2.0
Rate (/day)	10.0 ± 0.3	11.8 ± 0.6	11.5 ± 0.6
Neutron			
Thermal	0.03	0.04	---
Resonance	1.1	0.9	---
Total***	1.1	0.9	---
HIGH-LET**	82.8 ± 2.8	45.5 ± 2.3	21.1 ± 2.1
Total Mission Dose Equivalent**	166.3 ± 3.7	140.7 ± 5.3	56.5 ± 2.9
Mission Parameters			
Storage Locker	MF280	MF280	MF280
Duration (hrs)	29/19/148.5	192	73.6
Inclination (deg)	57.0	28.5	28.5
Altitude (km)	352/274/224	324	297-334

*Photons and electrons of any energies. High-LET at lower efficiency.
**HZE particles with LET > 20 keV/μm of water.
***Does not include high-energy neutron dose.

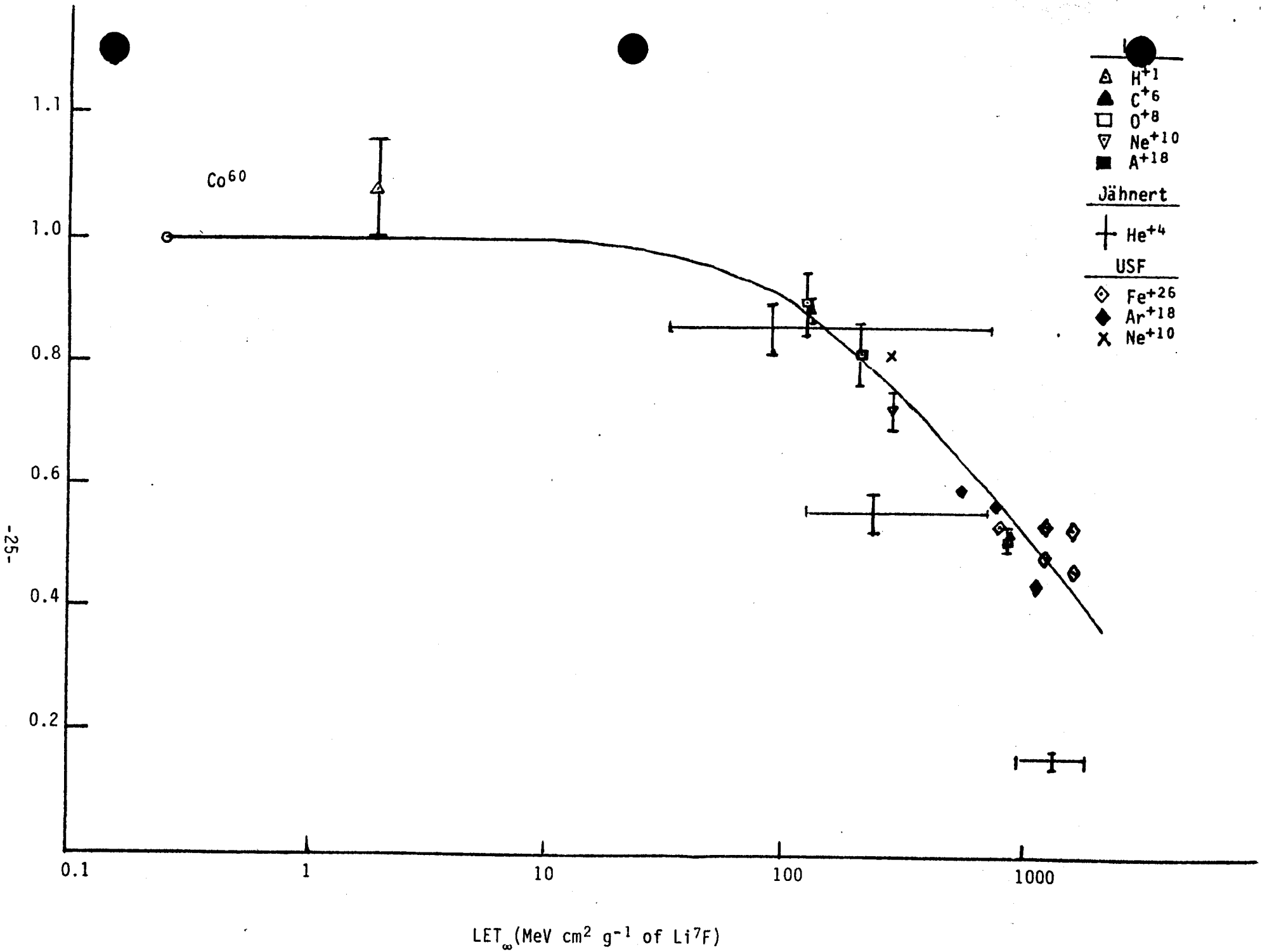


Fig. 7. Response of Li^7F as a function of LET. The response is expressed relative to cobalt-60 gamma rays.

matching pair of tracks where a heavy particle has intersected the interface. Background tracks and artifacts are therefore rejected during the scanning procedure. For all of the flights the CR-39 layers were processed by etching for 16 hrs. in 6.25N NaOH solution at 70°C.

Track densities of sizes above a pre-selected minimum size were counted on all the sample pairs. It is not desirable to include very small tracks where the counting efficiency is small, since reducing these track densities back to incident heavy particle fluences would be problematical. In order to get LET spectra from the tracks the minor axes of the track openings were measured with eyepiece grid micrometers or, in some cases, with electronic stage micrometers.

3. Effective LET cutoff of HZE particles

Of particular importance in deriving doses from the HZE measurements is the determination of the effective LET cutoff of the PNTD's. The acceptance criteria applied to track counting with the calibration relationship for CR-39,

$$\text{LET} = \exp \begin{cases} 2.341 + 0.1487\, x & x \leq -3.25 \\ 4.863 + 0.8954\, x - 0.1503\, x^2 - 0.01075\, x^3 + 0.01007\, x^4 & x > -3.25, \end{cases} \qquad (2)$$

where

$$x = \text{minimum } \{13, \ln[(Vt/Vb) - 1]\},$$

gives the dip angle dependent LET cutoff for the scanned layers, LET_c. In equation (2) above, V_t is the track etch rate, Vb the etch rate of the bulk material, and LET is the ionization rate of the particle.

An effective cutoff value can be determined from the dip angle dependent LET cutoff if two assumptions are made: (1) The particle beam is isotropic, and (2) the LET spectrum is given by a model spectrum. The model spectrum is taken from a computer-generated integral LET spectrum for a Skylab type of orbit. It

has been obtained from our colleague, W. Heinrich /12/. The typical effective shielding for the APD detector was estimated to be about 20 g/cm^2 of Al. Accordingly, the model integral LET spectrum is represented analytically by

$$F = C\ F_n, \tag{3}$$

where

$$F_n = \exp \begin{cases} 1.412 - 0.531\ x & x \leq -2 \\ 0.094 - 2.037x - 1.7151x^2 - 1.2446x^3 - 0.2994x^4 & -2 < x \leq 0 \\ -0.122 - 3.153x + 0.5617\ x^2 - 0.0875\ x^3 & 0 < x < 3.701 \\ -\infty & x \geq 3.701, \end{cases}$$

where

$$x = \ln(\text{LET}/134.7\ \text{keV}/\mu\text{m}), \tag{4}$$

LET = dE/dx in water, and C is a normalization constant, ultimately to be determined by the experimental fluence. F is the number of particles/cm^2-sr with energy loss rate > LET. The shape of the integral LET curve was obtained from the 20 g/cm^2 curve from W. Heinrich. The corresponding differential LET spectrum is given as the negative derivative of the integral LET spectrum,

$$f = -\ dF/d\text{LET} \tag{5}$$

The effective cutoff value of LET is computed using the definition that LET_c is the dip angle independent cutoff that would give the same fluence on a planar detector as the real dip angle dependent LET cutoff in an isotropic beam of particles distributed in LET as given by the model spectrum (Equation 3). For this computation the integral spectrum is represented as a power function in the vicinity of the LET cutoff of the plastic. From the model integral spectrum the exponent in this power function is found to be -1.149.

All of the LET values have been converted into the units of keV/μm of water with LET being $LET_\infty = dE/dx$. Actual calibration curves must be expressed as functions of LET_w if they are to accurately represent the results for all particles on a single curve. The subscript $_w$, which is of the order of 200 eV for CR-39, designates that LET_w only includes the energy loss to electrons receiving energies less than $_w$ in collisions with the stopping particle. The conversion factor from LET_w to LET_∞ is slightly Z-dependent. The value adopted is an average value of 1.462, taken to be applicable in the vicinity of the LET cutoff.

4. Model LET spectrum

Having obtained LET_c, the LET spectrum can be given by Equation (3), with C determined from the experimental fluence and LET_c. The measured fluence is on a planar detector, so the effective solid angle seen by the detector (for the effective LET cutoff) of 2π is the total solid angle of 4π steradian times the average projection factor of 1/2. Therefore, if F_p is the measured planar fluence, the equation

$$F_p = 2\pi\ F(LET_c) = 2\pi\ C\ Fn(LET_c) \qquad (6)$$

is used to determine C. The function $F_n(LET_c)$ is obtained from Equation (4).

5. Dose and dose-equivalent spectra

With an analytic expression for the LET spectrum the integral dose spectrum is obtained directly by integrating. The dose in millirads due to particles with LET > L is given by

$$D(LET>L) = 2.013E\text{-}3 \int_L^\infty LET\ f\ dLET \qquad (7)$$

The factor 2.013E-3 is the solid angle of 4π steradians times the conversion factor from LET units of keV/μm to millirads, 1.602E-4.

The integral dose equivalent spectrum is computed in exactly the same fashion as the dose spectrum except that an LET dependent weighting factor, called the quality factor, is included in the integration. The integral dose equivalent spectrum in millirem is given by

$$DE(LET>L) = 2.013E\text{-}3 \int_{L}^{\infty} LET\ QF\ f\ dLET \qquad (8)$$

The quality-factor is an interpolated form of the ICRP recommended QF(LET),

$$QF(LET) = \begin{cases} 1 & LET \leq 1.26 \\ 0.867\ LET^{0.616} & 1.26 < LET \leq 53 \\ LET^{0.58} & 53 < LET \leq 175 \\ 20 & LET > 175 \end{cases} \qquad (9)$$

where again LET is in keV/μm.

A comparison of a model and measured spectrum, from the STS-51A APD measurements, is shown in Fig. 8. Two measured LET spectra, from STS-51C, are given in Figs. 9 and 10. Fig. 9 shows the results of all coincident tracks found in the scanning while Fig. 10 shows only the spanning particle track results. Spanning particles are those traversing the full thickness of both CR-39 scanning sheets so that four aligned tracks are found in the surfaces of two sheets. The differences between the two spectra are due to stopping particles, whether primary or secondary in origin.

III. ACCELERATOR STUDIES

An extensive program of PNTD calibrations and studies of detector characteristics through accelerated beam irradiations is currently underway. Several ions (^{12}C, ^{16}O, ^{20}Ne, ^{40}Ar and ^{56}Fe) and beam energies have been employed to

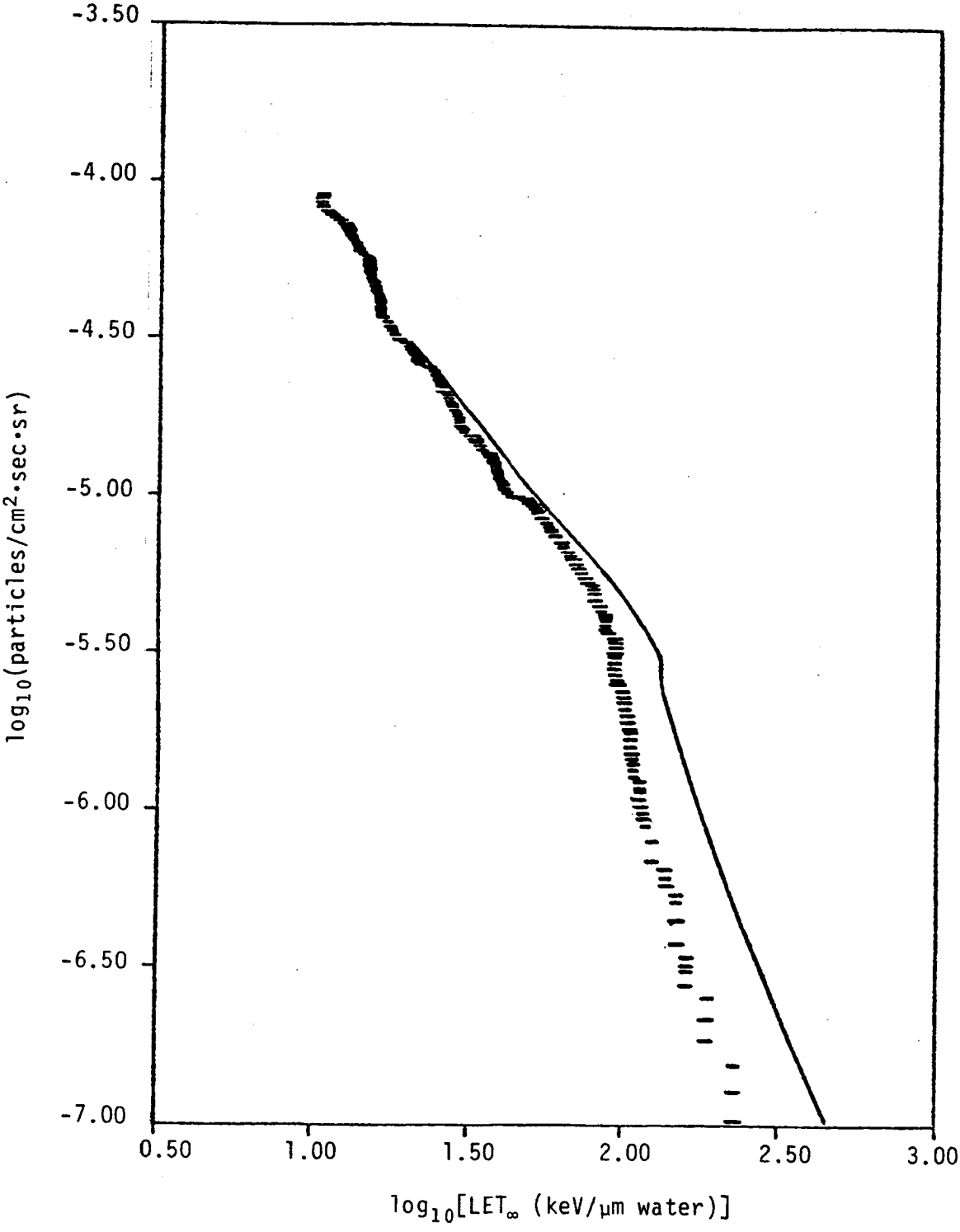


Fig. 8. Comparison of model and measured integral LET spectra on STS-51A.

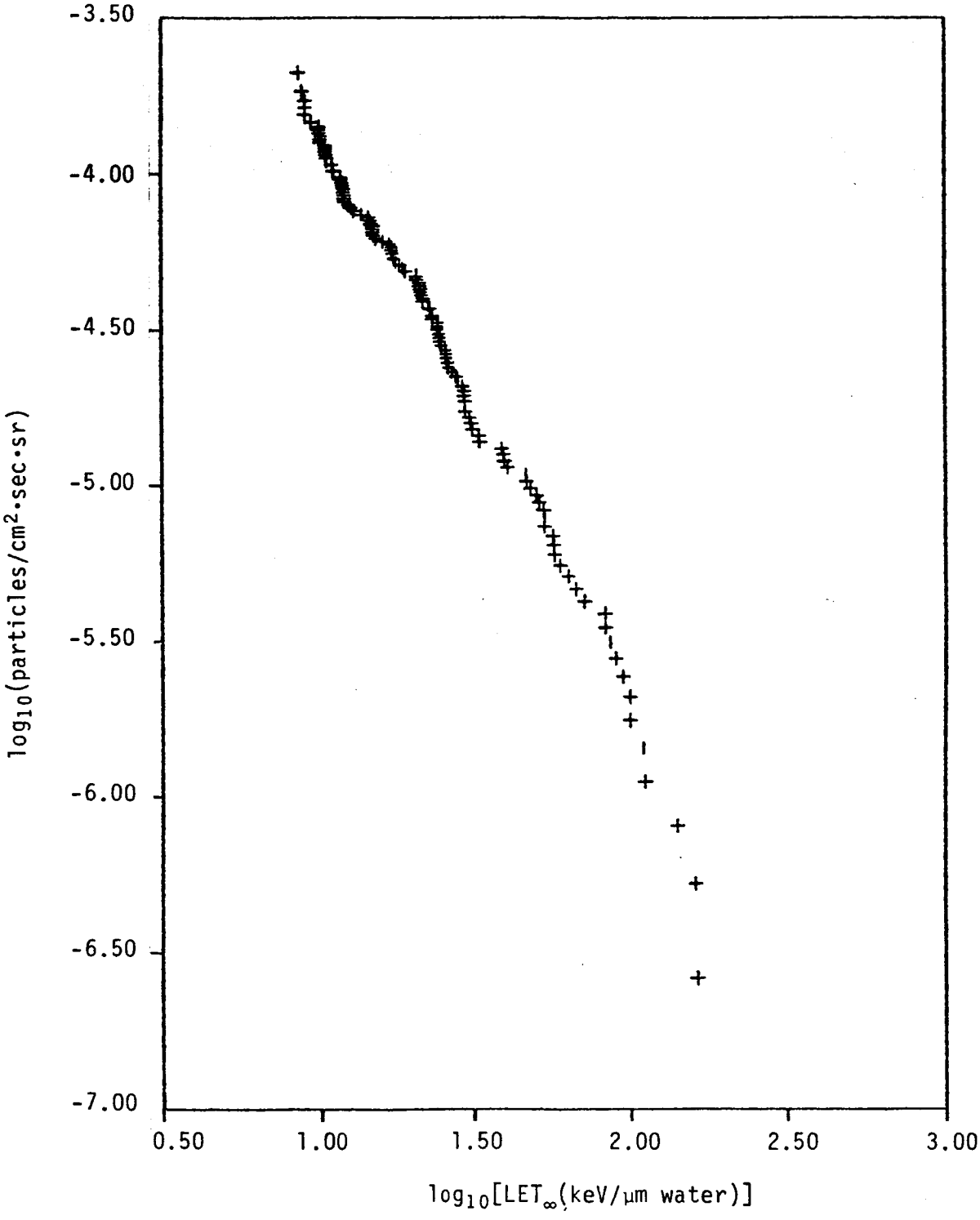


Fig. 9. Integral LET spectrum for STS-51C

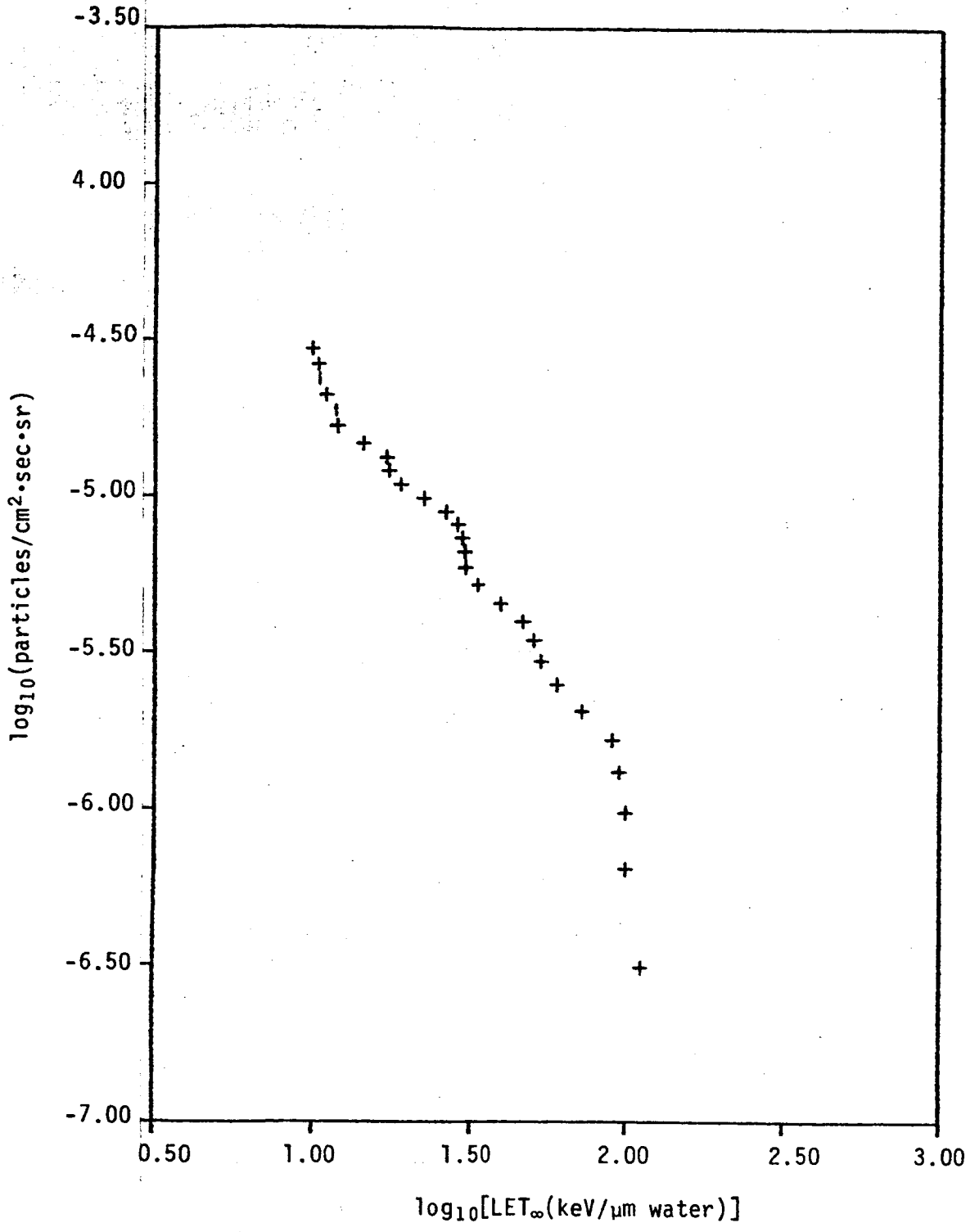


Fig. 10. Integral LET spectrum for spanning particles in STS-51C

provide a wide range of particle LET's This work will be completed and reported under the next contract. Also, dosimetry support at the Berkeley HILAC has been provided for other groups engaged in NASA Life-Sciences research projects.

IV. DOSIMETER DEVELOPMENT

In conjunction with the above accelerator studies, a program of dosimeter (PNTD) development is also under way. Repeated beam calibrations are needed to test the effects of different manufacturing formulas and methods of PNTD's. The results of these tests will be reported in the future but preliminary measurements show that CR-39, our most important PNTD, can be improved significantly over the types used in previous spaceflight work.

V. DOSIMETRY TECHNIQUES

One of the major objectives of this contract was the development of an automated analysis system for particle tracks in plastic track detectors. The system which was developed is described in Appendix 1. The appendix is arranged in the form of a complete report, since it represented a rather large fraction of the total effort.

REFERENCES

1. Benton E. V. and Henke R. P. (1983) Radiation exposures during spaceflight and their measurement. Adv. Space Res. 3, 171.

2. Janni J. (1969) A review of Soviet manned space flight dosimetry results. Aerospace Medicine 40, 1547.

3. Petrov V., Akatov Y., Kozlova S., Markelov V., Nesterov V., Redko V., Smirenny L., Khortsev A. and Chernikh I. (1975) The study of the radiation environment in near-Earth space. Space Res. 13, 129.

4. Bailey J. V. (1977) Dosimetry during space missions. IEEE Trans. Nucl. Sci., NS-23, #4, 1379.

5. Jordan, T. M. (1983) Radiation protection for manned space activities, JPL Publication 83-26, Jet Propulsion Laboratory, Pasadena.

6. Stassinopoulos, E. G. (1980) The geostationary radiation environment. J. Spacecraft and Rockets 17, 145.

7. Watts J. W. Jr. and Wright J. J. (1976) Charged particle radiation environment for the Spacelab and other missions in low Earth orbit--Revision A. NASA Tech. Memo. TMX-73358, 1-137.

8. Plato P. and Hudson G. (1980) Performance testing of personnel dosimetry service procedures manual. NUREG CR-1063, January 1980.

9. Roberts J. H., Parker R. A., Congel F. J., Kastner J. and Oltman B. G. (1970) Environmental neutron measurements with solid state track recorders. Rad. Effects 3, 283-285.

10. Benton E. V., Cassou R. M., Frank A. L. and Henke R. P. (1979) Space radiation dosimetry on board Cosmos 936: U.S. portion of Experiment K-206. Final Report, NASA Contract No. NAS2-9504.

11. Benton E. V., Henke R. P., Frank A. L., Johnson C., Cassou R., Tran M., and Etter E. (1980) Space radiation aboard Cosmos 1129: U. S. portion of Experiment K-309. Final Science Report, NASA Contract No. NAS2-10151.

12. Heinrich W. (1982) Private communication, University of Siegen, Siegen, Germany.

APPENDIX I

AUTOMATED ANALYSIS SYSTEM FOR HZE PARTICLE TRACK DOSIMETERS

TABLE OF CONTENTS

AUTOMATED ANALYSIS SYSTEM FOR HZE PARTICLE TRACK DOSIMETERS

1. INTRODUCTION

1.1 History of Automated HZE

The development of procedures for the automated readout of HZE (high Z and energy) detector stacks is the outgrowth of a long series of developments in HZE dosimetry and increasing requirements for rapid data retrieval. HZE dosimetric measurements have been made almost from the very beginning of manned spaceflight in the U.S. The majority of the general HZE measurements have been made by our group and several additional specialized measurements have been made by other groups using such esoteric dosimeters as the face plates of the astronauts' helmets. The earliest measurements were made using dosimeters called passive dosimetry packets. These were personnel dosimeters worn by the astronauts and consisted of two or three 8-12 cm^2 layers of Lexan polycarbonate plastic track detector. The principal dosimetric result from these detectors was a simple track count corrected for variations in sensitivity of the detectors. In addition, the LET spectrum was derived for the highest LET components, that is, those which are recorded in the relatively insensitive Lexan detectors. Also, very rough Z spectra were produced. These spectra were severely limited in both statistics and in the accuracy of the Z determination. These restrictions resulted from the small area and thinness of the detector stack.

From Apollo 14 on, the HZE dosimetry was considerably enhanced by the addition of a larger stack of detectors called the HZE dosimeter. This stack was 10 cm by 10 cm and typically 2 cm thick. Initially, it consisted of only Lexan and later on, in addition, cellulose nitrate detector layers. These

large stacks served as area monitors and one of them was included on each mission. Typically, they were placed on the minimum-shielding portion of the spacecraft wall. Because of its large thickness, the stack was able to capture the entirety of the registration range (portion of the particle trajectory in which registration occurs) of most of the particles stopping in the stack. This meant that quite accurate values of the Z of the particle were obtainable. Typically, the accuracy was better than ±1 in Z and ultimately the accuracy was refined to the point where isotopic resolution became possible. The main limitation of such stacks was the time required to measure and reduce the data they contained. Large readout times were necessary because hand measurements had to be made of the many tracks along the trajectories of the hundreds of particles stopping in each stack. Thus, man-months were required in order to reduce one of these stacks. Not only was such a procedure expensive, it also took much too long to produce the necessary dosimetric information from a mission. Therefore, the requirement for increased speed led very logically to what is now known as automated HZE.

1.2 Objectives

The objectives of the automated HZE program are 1) to be able to produce a complete record of the distribution of HZE particles stopping within or near a stack of plastic detectors; 2) to have the capability of obtaining this result very soon after the receipt of the dosimeters at the end of a mission. The first requirement places limits both on the minimum number of detector layers that must be used per stack and also on the minimum usuable area for each of these layers. In addition, the dosimetric information must be complete. This stems largely from the fact that the biological effects of HZE radiation are poorly understood at present. Therefore, the HZE dosimeters must measure all of the relevant physical data about the radiation field encountered by the astronauts. The second requirement places limits on the nature of the

physical measurements that are made on the detectors and the speed at which these measurements are made. Obviously, the previous approach of hand measurements is impossible if this second requirement is to be met.

The general area of automated-HZE analysis also includes the automatic scanning and measurement of individual layers of etched plastic, that is, those not necessarily taken from a large stack. In these cases, the data are not analyzed using the full complement of auto-HZE programs but are often used directly to obtain a statistical result, such as a particle fluence or an LET spectrum.

1.3 General Approach

To meet the above stated objectives, it was decided that the processed detectors would be analyzed via an optical digitizer. To make this automated readout possible special detector processing techniques have been developed. The information is processed in a computer and the final results are produced, again by the computer. Using this concept, a minimum amount of labor is involved and a maximum amount of speed is achieved.

1.4 Summary of Components

The components can logically be divided into two categories, hardware and software. The first hardware component is an optical digitizer. Since the required resolution demands that the field of view be much smaller than the area of the detector, an automated stepping stage is also required to scan the sample. To both control the process and receive the data, a computer is required. The same computer is also used for processing the data which has been received. In addition, it is necessary to have such minor pieces of equipment as interfaces and several computer peripherals necessary for storing the information and for input and output operations to the computer.

The software component consists of the several programs necessary to

take the data and reduce it into the final output form. In addition, software is required for establishing the values of system parameters, testing the system, testing the concepts, and verifying the validity of the results.

2. SYSTEM REQUIREMENTS

The principal requirement and the main motivation for automated HZE analysis is that the detector stacks be read out very quickly after they have been processed. This requirement translates into a reduction in the cost of analysis since the main cost is the time spent by the personnel in measuring and reducing the data.

The second requirement is for hardware simplicity. It stems from the desire to reduce cost and minimize the chances of hardware failure.

The third requirement is based on the character of the measurements necessary to produce the intended output. The final output will be statistical distributions of track parameters such as the atomic number, Z, or the LET of the particle. To obtain such parameters for individual particles, it is necessary to make several measurements along the trajectory. First of all, the positions must be measured and traced from layer to layer to obtain the trajectories of the particle. In addition, some other parameter which varies monotonically with the particle LET must be measured. For this study, we have chosen this parameter to be the track-opening area. This is the projected area of the etched-through track obtained after a very long etch.

The first statistical distribution of interest is the LET spectrum. This is the distribution of the particle LET values. These can be obtained with a relatively thin stack of plastic. The second distribution of interest is the particle Z spectrum. In addition to the distribution in Z, however, it is necessary to know some other parameter such as the particle energy. Thus the Z spectrum is actually the distribution in Z and energy.

3. OVERALL SYSTEM DESIGN

3.1 Basic Concepts

The basic concept behind the HZE dosimetry system involves the use of a stack of mixed detector types. These will be alternating detector layers of differing sensitivities. Thus it will be possible to span the entire range of LET values measurable with solid-state nuclear track detectors.

The detectors will be given a very long etch. The purpose of the long etch is to provide etched-through holes that are very large. During this etch, approximately 0.6 to 0.9 of the detector material will be removed. By providing very large etched-through holes, a very contrasty feature will be availfor automated optical analysis.

The second step of the process consists of digitizing the layers. This will be accomplished by an optical digitizer which is interfaced to a computer. The data received from each portion on the layer will be analyzed to yield the track positions and areas for each of the etched-through holes.

The third step is to trace each of the particles through the stack, that is, to find those holes which lie along each particle trajectory and are, therefore, correlated in their relative position in space. In this step, two things are accomplished. Firstly, the particle trajectories are determined; secondly, the data is reorganized from the form of being ordered by layer to the form of being ordered by individual particle.

The fourth step is to calculate the remaining two parameters for each particle. These two parameters are its identity (value of Z) and stopping point. It is necessary to calculate the stopping point of the particle since it is not precisely measured. The position can only be inferred from the raw measurements by noting the point at which the track measurements cease along the trajectory.

The fifth and final step in the reduction procedure is to collect the data about the individual particles into statistical distributions. This step then produces a reduced volume of data that can be analyzed in terms of the biological hazard introduced. Of course, inherent in the biological analysis is the requirement that information relating biological damage to such parameters as particle LET be available.

3.2 Implementation Procedure

The above-mentioned concepts will be implemented as follows. The overall stack design will consist of an orthogonal set of detector stacks. The orthogonality of the stacks is necessary because of the anisotropy of the radiation field. Since the detectors are more sensitive along the normal to the detector surface, it is necessary to have stacks oriented in the three orthogonal directions. In this way, the final statistical analyses will involve an averaging process in which the results from the different stacks are combined.

Each of the stacks will consist of sequences of the following types of material: CR-39, Lexan polycarbonate and Kodak-Pathé cellulose nitrate. Currently it is thought that the best sequence of layers is CR-39, Lexan, CN, Lexan, CR-39, Lexan, etc. The CR-39 and CN layers are the most sensitive and will record the low-LET portion of the particle spectrum. The Lexan layers are less sensitive and, therefore, are capable of measuring the LET even for relatively high LETs, such as those encountered near the stopping point of iron particles.

Stacks with the specific objective of monitoring personnel exposure will be a subset of the above mentioned stacks. Because of size and weight requirements, they will contain only a few layers which will not be arranged in an orthogonal configuration. Such stacks will contain layers representative of the different detector types with possible duplications for redundancy and use with different processing conditions.

After having been exposed, the detectors will be processed by etching them in heated NaOH solutions. The minimum final thickness for each of the layers will be approximately 40 μm. The minimum thickness is determined by the requirement that spurious measurement of very short-range recoil-type particles be excluded. It is desired, however, that the maximum amount of material possible be removed to increase the sensitivity of each of the detectors.

The detectors will be read out using a Eyecom image digitizer made by Spatial Data Systems in Goleta, California. This piece of hardware is directly connected to a PDP-11/34 computer system. It resides on the bus of the system and, therefore, is programmed directly as any standard DEC peripheral would be programmed. The detector layers will be digitized with what is called the high-speed digitizer of the Eyecom. For each field digitized, a 240 by 320 pixel array of gray levels is placed in the refresh memory of the Eyecom. Each of these picture elements has an accuracy of five bits or 1 part in 32 in gray level. Then the stage will be translated and the information in the Eyecom refresh memory will be partially processed and then transferred to disk storage. Subsequently, the stage will be moved in a raster pattern to completely cover the area of interest. This whole process will be repeated for each layer of the stack.

Following the reading of the data, it will be processed on the 11/34 to produce the final results which were described above and which will be described in much greater detail in the following sections.

4. THEORETICAL FOUNDATION

This section describes the theoretical foundation of several of the automated HZE data-reduction steps. For the reader unfamiliar with least-squares theory, some or all of the "Theoretical Foundation" section can be treated as an appendix. A qualitative description of the algorithms used in the computer codes is given in the section on "software".

4.1 Least-Squares Particle Trajectories

Since the particle's trajectories are straight lines in space, we can express both the x and y positions of the particle tracks as linear functions of z,

$$x = a_{1x} + a_{2x}z \tag{4-1}$$

$$y = a_{1y} + a_{2y}z \tag{4-2}$$

The point (x,y) is the lateral position of the particle trajectory as it intersects the center of the detector layer and z is the physical height of this layer center (in mm for example) from some arbitrary origin (usually the bottom of the stack). The relationship of the parameters a_{2x} and a_{2y} to the familiar trajectory azimuthal angle, α, and the dip angle, δ, can be seen in Fig. 4.1.

$$a_{2x} = \frac{\partial x}{\partial z} = \text{ctn } \delta \cos \alpha \tag{4-3}$$

$$a_{2y} = \frac{\partial y}{\partial z} = \text{ctn } \delta \sin \alpha \tag{4-4}$$

Since at the initial stage of the data reduction the direction of particle

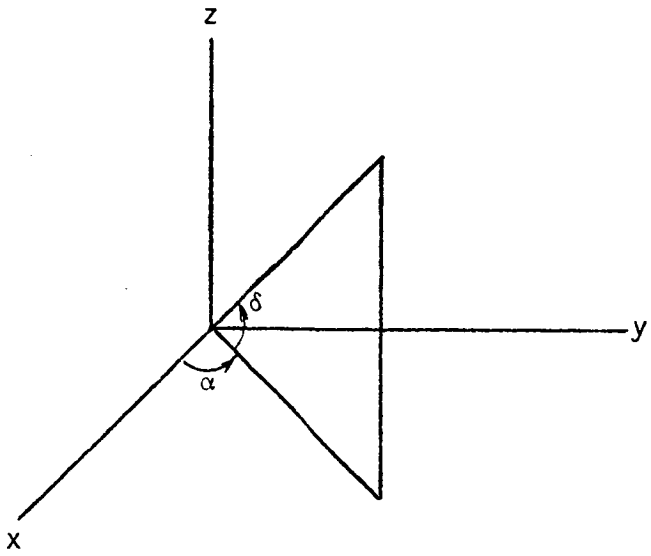


Figure 4.1. Trajectory angle definition.

travel along the trajectory is unknown; only the positive z hemisphere needs to be considered in the trajectory angle definition. That is, $0 \leq \alpha \leq 2\pi$ and $0 \leq \delta \leq \frac{\pi}{2}$. Therefore α and δ are obtained from a_{2x} and a_{2y}

$$\tan \alpha = \frac{a_{2y}}{a_{2x}} \tag{4-5}$$

and

$$\tan \delta = \frac{1}{\sqrt{a_{2x}^2 + a_{2y}^2}} \tag{4-6}$$

The quandrant for α is given in Table 4.1.

TABLE 4.1. QUADRANT FOR α

	$\tan \alpha < 0$	$\tan \alpha > 0$
$a_{2x} < 0$	2	3
$a_{2y} > 0$	4	1

The values of a_{1x}, a_{2x}, a_{1y}, and a_{2y} are obtained through a linear regression for the measured values of x and y, x_m and y_m, as a function of z. The different values of z are obtained, of course, on the different layers of the stack. For a thorough explanation of least-squares theory, see Orear (1958).

The solution is obtained by minimizing the sums of the residuals squared,

$$M_x = [(x - x_m)^2] \tag{4-7}$$

and

$$M_y = [(y - y_m)^2] \tag{4-8}$$

with respect to a_{1x}, a_{2x}, a_{1y}, and a_{2y}. Here the computed values of x and y are obtained from Eqs. (4-1) and (4-2). The bracket notation, which is somewhat standard in least-squares theory, is defined by

$$[q] \equiv \sum_{\text{data points}} q_i \tag{4-9}$$

where q_i is the value of q for the ith measurement. The solution is obtained by setting the partial derivatives of M with respect to the unknown parameters equal to zero. In matrix form, the solution can be expressed as

$$\langle \underline{a}_x \rangle = \begin{pmatrix} a_{1x} \\ a_{2x} \end{pmatrix} = \underline{\underline{H}}^{-1} \cdot \underline{b}_x \tag{4-10}$$

$$\langle \underline{a}_y \rangle = \underline{\underline{H}}^{-1} \cdot \underline{b}_y \tag{4-11}$$

where

$$\underline{\underline{H}} = \frac{1}{2} \begin{pmatrix} \dfrac{\partial^2 M}{\partial a_1^2} & \dfrac{\partial^2 M}{\partial a_1 \, \partial a_2} \\ \dfrac{\partial^2 M}{\partial a_1 \, \partial a_2} & \dfrac{\partial^2 M}{\partial a_2^2} \end{pmatrix} = \begin{pmatrix} [1] & [z] \\ [z] & [z^2] \end{pmatrix} \tag{4-12}$$

$$\underline{b}_x = \begin{pmatrix} [x_m] \\ [x_m z] \end{pmatrix} \tag{4-13}$$

$$\underline{b}_y = \begin{pmatrix} [y_m] \\ [y_m z] \end{pmatrix} \tag{4-14}$$

The angle bracket notation, $\langle\ \rangle$, indicates "expected value of". An underline indicates a vector and a double underline indicates a matrix.

The spread of the data about the fit lines is given by

$$\sigma_m^2 = \frac{[(x - x_m)^2]}{[1] - 2} = \frac{[x_m^2] - \langle \underline{a}_x \rangle \cdot \underline{b}_x}{[1] - 2} \tag{4-15}$$

and

$$\sigma_m^2 = \frac{[(y - y_m)^2]}{[1] - 2} = \frac{[y_m^2] - \langle \underline{a}_y \rangle \cdot \underline{b}_y}{[1] - 2} \tag{4-16}$$

The variances in x and y given in Eqs. (4-15) and (4-16) are obtained by assuming that chi squared for the least-squares fit is equal to the number of degrees of freedom, $[1] - 2$, which appears as the denominator. Then the covariance matrices for $\langle \underline{a}_x \rangle$ and $\langle \underline{a}_y \rangle$ are given by

$$\left\langle \left[a_{ix} - \langle a_{ix} \rangle \right] \left[a_{jx} - \langle a_{jx} \rangle \right] \right\rangle = \sigma_x^2 H_{ij}^{-1} \tag{4-17}$$

$$\left\langle \left[a_{iy} - \langle a_{iy} \rangle \right] \left[a_{jy} - \langle a_{jy} \rangle \right] \right\rangle = \sigma_y^2 H_{ij}^{-1} \tag{4-18}$$

Two separate covariance matrices are given for x and y because the errors in x and y are totally uncorrelated. With the expression for the variances in a_{2x} and a_{2y}, we can now obtain the error in dip angle through the standard error propagation formula using the partial derivatives of Eq. (4-6) with respect to a_{2x} and a_{2y}.

$$\sigma_\delta^2 = \tan^2\delta \sin^4\delta \left(\langle a_{2x} \rangle^2 \sigma_x^2 + \langle a_{2y} \rangle^2 \sigma_y^2 \right) H_{22}^{-1} \tag{4-19}$$

4.2 Projected Etch-Through Areas

Because the computation of the etch-through areas of tracks is quite involved, it will be treated separately in this section. The complexity of the computation arises because the intersection of up to five different ellipses is involved in the periphery of the projected track opening.

Figure 4.2 shows the parameters of a track in a layer etched to remove 0.9 of the original layer thickness. The particle dip angle, δ, is 60 deg and the track cone angle, θ, is 45 deg. Part a of the figure demonstrates the track as seen from the side, and part b shows it as seen from above. The amount of material removed during the etch is B and the original layer thickness is h_0. The fraction of the layer removed is $\rho = \frac{2B}{h_0}$, which can also be expressed as the value of B normalized to $\frac{h_0}{2}$, the half-layer thickness. In the ensuing discussion all physical dimensions will be normalized to $\frac{h_0}{2}$ to considerably simplify the expressions. In the normalized units, $B \rightarrow \rho$ and $h_0 \rightarrow 2$.

The parameters of the projected ellipses are shown in Fig. 4.2b. The values of a_1 and b_1 are obtained by considering Fig. 4.2a. From Henke and Benton (1971) the value of L (normalized) is $\frac{\rho}{\sin\theta}$. The distance along the trajectory from the pre-etch surface to the central intersection circle is $\frac{1}{\sin\delta}$. Therefore b_1, which is the radius of this circle, is given by

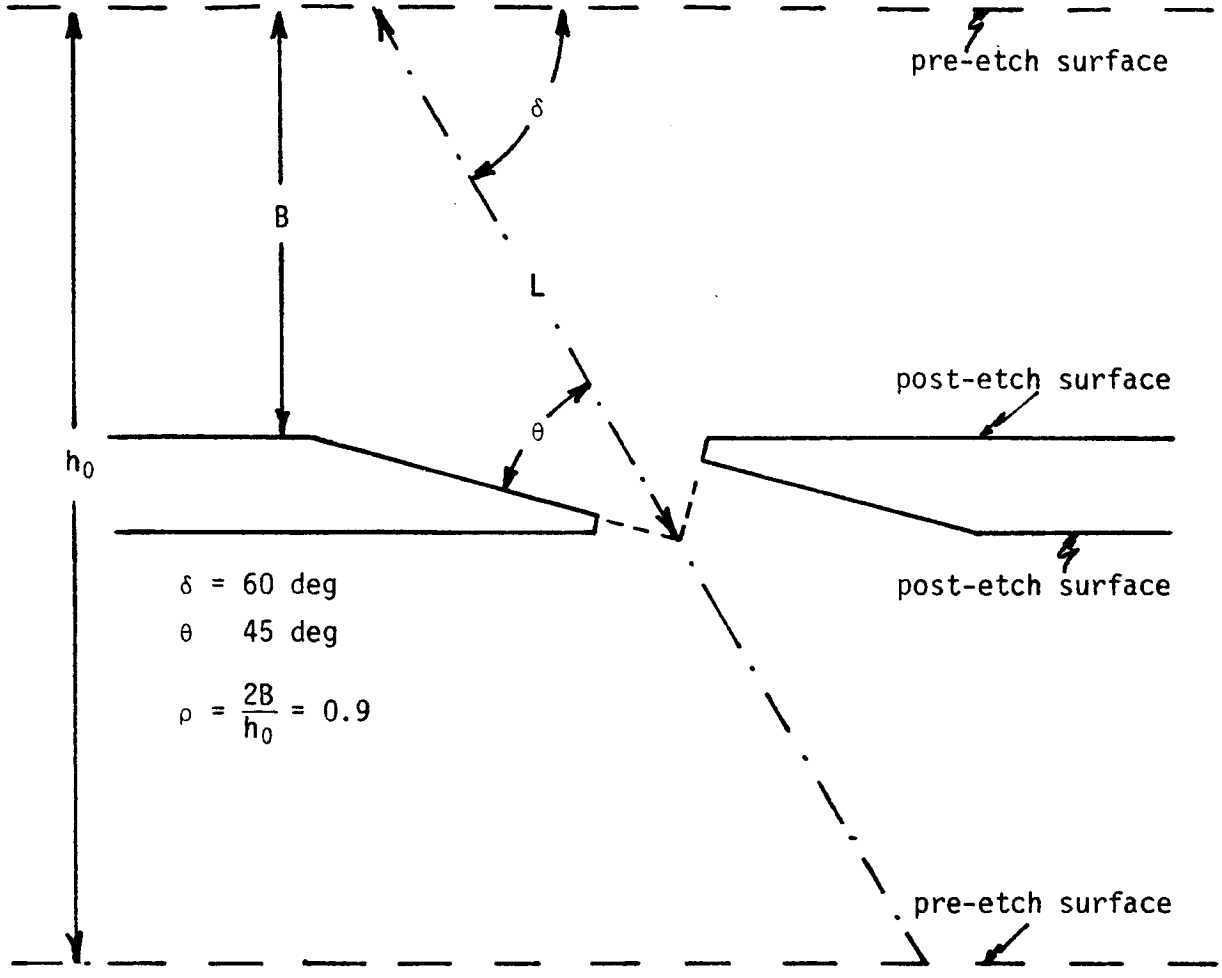
$$b_1 = \left(\frac{\rho}{\sin\theta} - \frac{1}{\sin\delta}\right)\tan\theta = \frac{\rho}{\cos\theta} - \frac{\tan\theta}{\sin\delta} \qquad (4\text{-}20)$$

The projection of b_1 onto the horizontal plane is a_1, given by

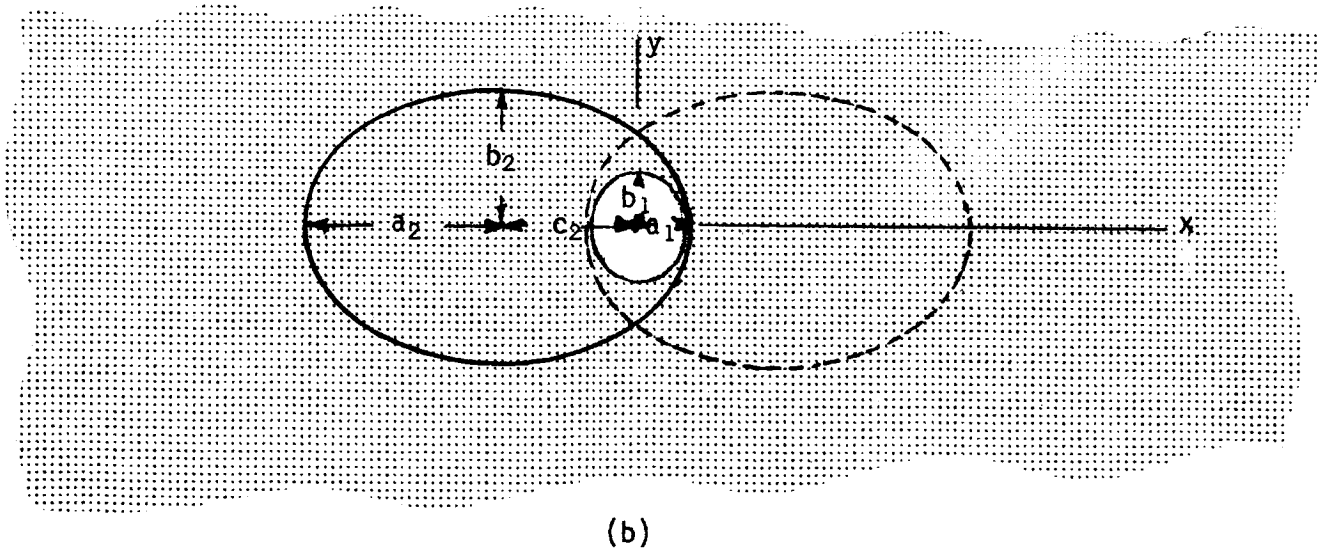
$$a_1 = b_1 \sin\delta = \frac{\rho \sin\delta}{\cos\theta} - \tan\theta \qquad (4\text{-}21)$$

Since this ellipse is centered on the defined x,y coordinate system, c_1 (not shown), the distance of the ellipse center from the origin is zero.

For the surface ellipse, the values of a and b as given by Henke and Benton (1971) are


pre-etch surface
δ
B
L
θ
post-etch surface
h_0
δ = 60 deg
θ 45 deg
$\rho = \frac{2B}{h_0} = 0.9$
post-etch surface
pre-etch surface

(a)


y
b_2
b_1
a_2
c_2
a_1
x

(b)

Figure 4.2.

$$a_2 = \frac{\rho \cos \theta}{\sin \delta + \sin \theta} \tag{4-22}$$

and

$$b_2 = \rho\sqrt{\frac{\sin \delta - \sin \theta}{\sin \delta + \sin \theta}} \quad . \tag{4-23}$$

From the geometry in Fig. 4.2a and Henke & Benton (1971), p. 489, the displacement of the center of the ellipse from the coordinate system origin is

$$c_2 = \frac{1 - \rho}{\tan \delta} + \frac{a_2 \tan \theta}{\tan \delta} = \frac{1}{\tan \delta} - \frac{\rho \cos \delta}{\sin \delta + \sin \theta} \tag{4-24}$$

For the special case geometry shown in Fig. 4.2. it is easy to compute the projected track opening area, $A_p = \pi a_1 b_1$ and the perimeter, $P_p = 4b_1 E(\cos \delta) \approx 2\pi\sqrt{\frac{a_1^2 + b_1^2}{2}}$, (Hodgman, 1950) because the projected opening is delimited by the central ellipse alone. Here E is the complete elliptic integral of the second kind. In the more general case, however, the surface tracks may be undercut $\left(\delta + \theta < \frac{\pi}{2}\right)$ and the projected hole opening will either be partially or wholly delimited by the surface ellipses.

Another complexity that can arise is that the upper surface of the post-etch layer may actually intersect the cone produced from the lower surface etch and vice versa. This situation occurs for a long etch, i.e.

$$\rho > \frac{\tan \delta \cos \theta + \sin \theta}{\tan \delta \cos \theta + \sin \delta} \tag{4-25}$$

when the surfaces have etched down through the central circle. We will call this ellipse formed by the intersection of the other-surface cone with the layer surface the "other-surface ellipse". The parameters a (semi-major axis), b (semi-minor axis), and c (displacement of the center from the coordinate system origin) are designated by a_3, b_3, and c_3 for the other-surface ellipse. It can be shown by considerations similar to those used in deriving the previous ellipse parameters that a_3, b_3, and c_3 are given by

$$a_3 = \frac{\rho \cos\theta}{\sin\delta - \sin\theta} - \frac{\sin 2\theta}{\sin^2\delta - \sin^2\theta} , \qquad (4\text{-}26)$$

$$b_3 = \rho\sqrt{\frac{\sin\delta + \sin\theta}{\sin\delta - \sin\theta}} - \frac{2\sin\theta}{\sqrt{\sin^2\delta - \sin^2\theta}} , \qquad (4\text{-}27)$$

and

$$c_3 = \frac{\operatorname{ctn}\delta}{\sin\delta - \sin\theta}\left(\frac{\sin^2\delta + \sin^2\theta}{\sin\delta + \sin\theta} - \rho\sin\delta\right) . \qquad (4\text{-}28)$$

We now have all of the parameters for the three different types of ellipses that can be involved in the delimiting of the perimeter of the projected opening for the etched-through track. They are given by Eqs. (4-20) (4-21), and $c_1 = 0$ for the central ellipse, Eqs. (4-22) to (4-24) for the surface ellipses, and Eqs. (4-26) to (4-28) for the other-surface ellipses. For the surface and other-surface ellipses, a denotes the semi-major axis and b denotes the semi-minor axis. For the central ellipse, the roles are interchanged to allow a to represent the x direction extension and b the y direction extension of the ellipse. In each case, c is the displacement of the ellipse center from the origin. There are two surface ellipses centered at $x = \pm c_2$, and when they occur, two other-surface ellipses centered at $x = \pm c_3$.

Although the above mentioned 5 ellipses can each participate in the framing of the projected opening, the calculation of the opening area and perimeter can be reduced to the involvement of only two ellipses for a given set of δ and θ. This is accomplished by utilizing the natural symmetry in y and the symmetry in x produced by the assumption that θ is constant throughout the layer, making the cones on the two surfaces identical. Therefore, we need consider only one quadrant of the coordinate system and the results for this quadrant will be multiplied by four. It should also be noted that the surface and other-surface ellipses are not simultaneously involved because the other-surface ellipse replaces the surface ellipse when inequality (4-25)

is satisfied.

Most of the configurations of etched-through tracks are shown in Figs. 4.3 and 4.4. Figure 4.3 shows non-undercut tracks $\left(\delta + \theta > \frac{\pi}{2}\right)$ and Fig. 4.4 shows undercut tracks $\left(\delta + \theta < \frac{\pi}{2}\right)$. In part (a) of the figures, the side view of the plastic is shown for two different etch times. The smaller etch time is sufficiently long to have etched the track through but the surface has not encountered the central circle. In Fig. 4.3b the central ellipse alone determines the boundary of the projected etched-through hole. In Fig. 4.4b both the central and the surface ellipses are involved. Another case which is not shown in Fig. 4.4 is for a slightly shorter etch than the one shown in Fig. 4.4b. In this case, the two surface ellipses intersect at a y value which is less than b_1 and the two surface ellipses delimit the projected hole. The character of the tracks produced in highly etched layers is shown in Figs. 4.3c and 4.4c. In these cases, the central and the other-surface ellipses delimit the projected hole.

The projected track opening area and perimeters are computed by combining the contributions from the segments due to each of the ellipses involved. For this purpose, it is necessary to find the intersection point of the ellipses involved. The intersection point for ellipse i with ellipse j can be obtained by noting the ellipse equation,

$$\frac{(x - c_i)^2}{a_i^2} + \frac{y^2}{b_i^2} = 1 \quad , \tag{4-29}$$

and then equating the y^2 values for the ith and jth ellipses

$$b_i^2\left[1 - \frac{(x - c_i)^2}{a_i^2}\right] = b_j^2\left[1 - \frac{(x - c_j)^2}{a_j^2}\right] \tag{4-30}$$

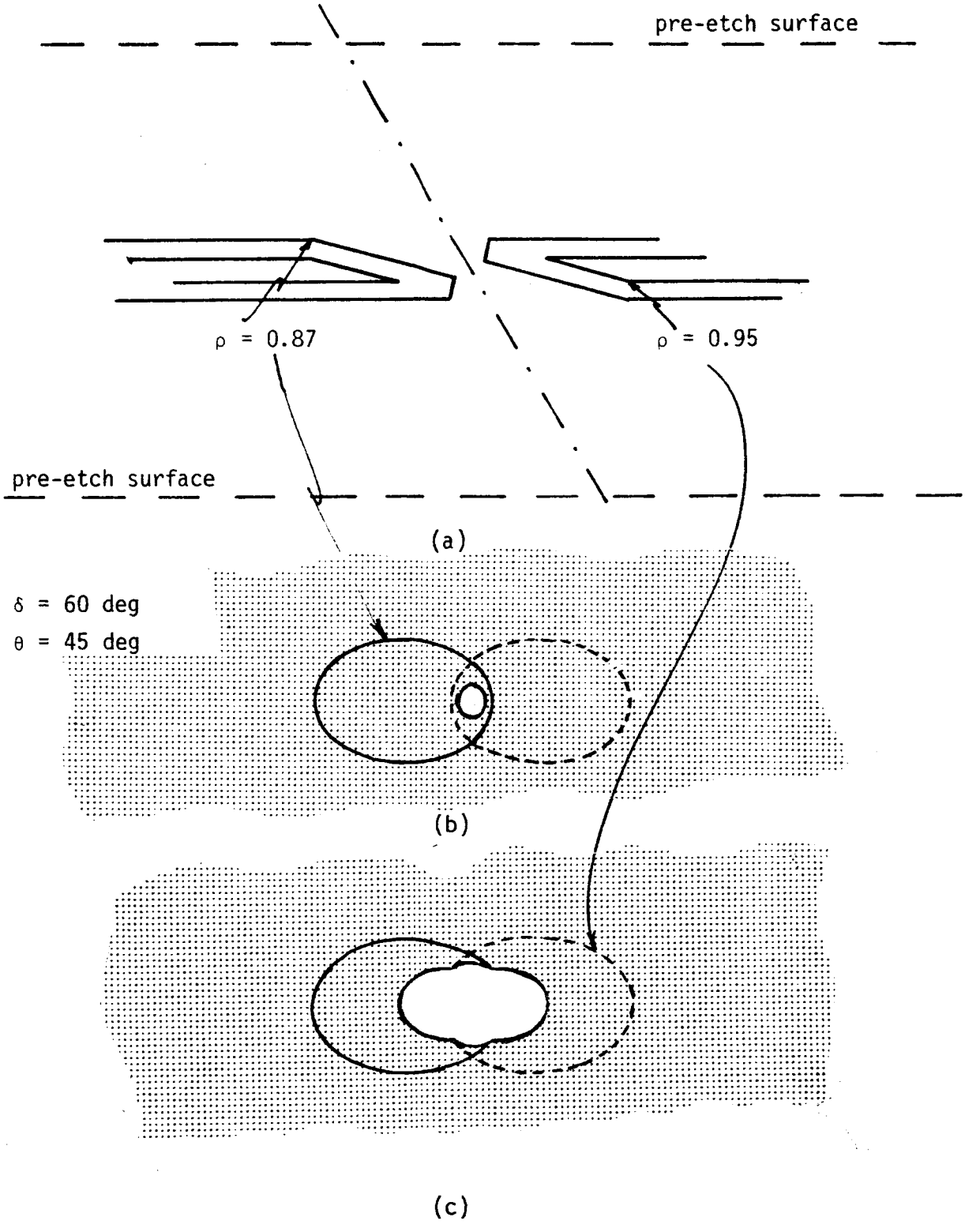


Figure 4.3. Non-undercut tracks.

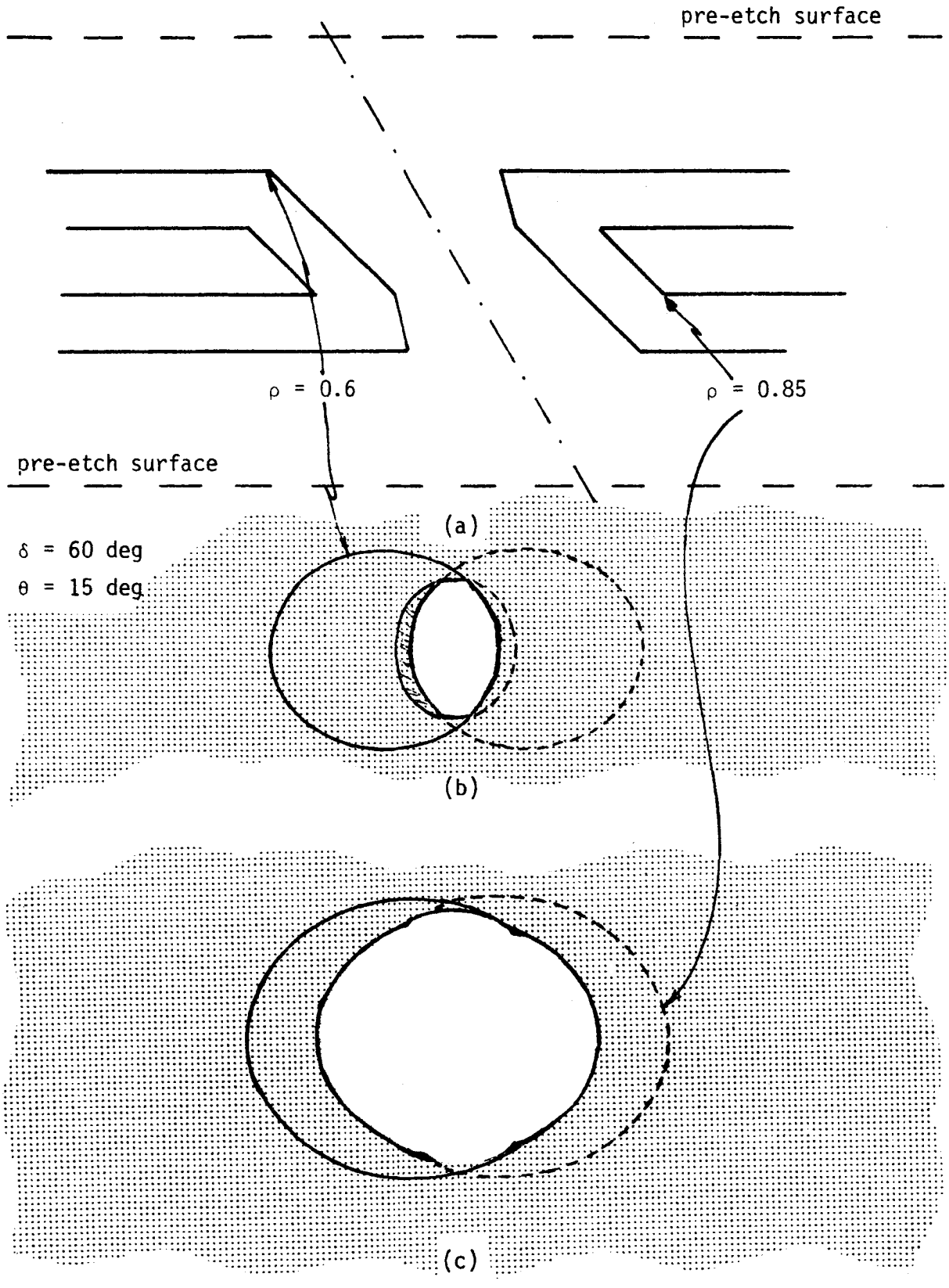


Figure 4.4. Undercut tracks.

Equation (4-30) leads to the quadratic equation

$$\left(\frac{b_i^2}{a_i^2} - \frac{b_j^2}{a_j^2}\right)x^2 - 2\left(\frac{b_i^2 c_i}{a_i^2} - \frac{b_j^2 c_j}{a_j^2}\right)x + b_i^2\left(\frac{c_i^2}{a_i^2} - 1\right) - b_j^2\left(\frac{c_j^2}{a_j^2} - 1\right) = 0 \qquad (4\text{-}31)$$

for the x value of the intersection point of the two ellipses. Here it is assumed that the values of c_i and c_j have the proper sign, depending on whether they are produced by the upper or lower surface intersection. The previous expressions for the c_i have only given their magnitudes.

The area and perimeter contribution from the 1st quadrant section of ellipse i lying between the bounding x values, x_1 and x_2, are given by

$$\Delta A_i = \int_{x_1}^{x_2} y \, dx \qquad (4\text{-}32)$$

and

$$\Delta P_i = \int_{x_1}^{x_2} \sqrt{dx^2 + dy^2} = \int_{x_1}^{x_2} \sqrt{1 + (dy/dx)^2}\, dx \,, \qquad (4\text{-}33)$$

respectively. The values of y and $\frac{dy}{dx}$ are given by

$$y = b_i\sqrt{1 - \left(\frac{x - c_i}{a_i}\right)^2} \,, \qquad (4\text{-}34)$$

$$\frac{dy}{dx} = -\frac{b_i(x - c_i)}{a_i\sqrt{a_i^2 - (x - c_i)^2}} \,, \qquad (4\text{-}35)$$

and it is again assumed that the c_i are appropriately signed values. With the transformations

$$u = \frac{x - c_i}{a_i} \,,$$

$$\phi = \sin^{-1} u \,,$$

and

$$k^2 = 1 - \left(\frac{b_i}{a_i}\right)^2$$

Equations (4-32) and (4-33) can be put in the form

$$\Delta A_i = a_i b_i \int_{u_1}^{u_2} \sqrt{1 - u^2}\, du \qquad (4\text{-}36)$$

and

$$\Delta P_i = a_i \int_{\sin^{-1}u_1}^{\sin^{-1}u_2} \sqrt{1 - k^2 \sin^2\phi}\, d\phi \qquad (4\text{-}37)$$

where $u_1 = u(x_1)$ and $u_2 = u(x_2)$.

The integral in Eq. (4-36) can be obtained in closed form and is given by

$$\Delta A = \tfrac{1}{2} a_i b_i (u_2\sqrt{1 - u_2^2} - u_1\sqrt{1 - u_1^2} + \sin^{-1}u_2 - \sin^{-1}u_1)$$

The integral in Eq. (4-37) is an incomplete elliptic integral of the 2nd kind and as such it is not easily expressable as an analytic function. Therefore, the integration is performed numerically in the program using a three point gaussian quadrature for each of the ellipse sections.

The total area and perimeter is then given by

$$A = 4 \sum_{\text{allowed } i} \Delta A_i \qquad (4\text{-}38)$$

and

$$P = 4 \sum_{\text{allowed } i} \Delta P_i \quad . \qquad (4\text{-}39)$$

The factor of 4 allows for the contribution due to all four quadrants and the allowed values of i are those indicating ellipses involved in the periphery of the projected hole.

4.3 Z-Analysis Least-Squares Theory

For each particle which intersects the detector stack with sufficiently large values of LET to produce etch-through tracks, one or more measurements of the projected track opening area are available. At the Z-analysis stage, the track position measurements have already been analyzed both to correlate the track area measurement for each particle which registers in three of more layers and to determine the trajectory parameters for the particle. Therefore, the data received by the Z-analysis program are the several projected area measurements and the relative positions along the particle trajectory at which each of these measurements was made. In addition, we know the particle dip angle, which is a necessary component of the measurement set. The particle stopping point is not known, however. Its approximate location can possibly be determined by the disappearance of tracks as one proceeds along the trajectory in the direction of increasing LET. At best this determination has a rounding error of ½ layer and moreover since the stack contains an arbitrary layer configuration in the general case, it is possible that an inert or insufficiently sensitive layer or possibly even a defect occurs at the terminal end of the trajectory. Therefore, the particle stopping point is taken to be unknown but is constrained to lie outside of the set of measured areas. Thus we have as unknowns the particle stopping point and its identity.

The particle identity consists of the parameter set Z and M, the atomic number and mass of the particle, respectively. It is overpresumptuous of the data accuracy to expect that isotopic resolution can be achieved. We therefore replace the set Z,M by a single parameter, the effective Z, which for convenience we also label Z. It is treated as a continuous variable. It should be remembered, however, that the effective Z is not strictly the particle atomic number but also reflects the mass of the particle, with 5-10 mass units being equivalent to one unit in Z. Therefore, we do not expect the effective Z to lie precisely at integral numbers because the transformation between

Z and A and Z effective assumes (see the next paragraph) that the most expected value of M varies smoothly with Z, which is not the case for the real distribution of isotopes. The transformation between Z and M and the effective Z is accomplished by replacing M in the formulation with a function M(Z) which approximates the most abundant isotope for the element. A number of functional forms are possible to represent M(Z) to the required accuracy. The one chosen is a power function because it is most consistent with the rest of the formulation. The actual representation used is

$$M = M_1 Z^{\xi} \quad , \tag{4-40}$$

where $M_1 = 1.776$ and $\xi = 1.059$. Equation (4-40) is a least-squares fit to the most abundant isotope of the ions $6 \leq Z \leq 26$. It also represents the most abundant isotopes of the elements adjacent to this range quite well.

The basic objective in the least-squares analysis for Z is to minimize the sums of the squares of the residuals, M, with respect to Z and the particle stopping point. With

$$M = \sum_{\text{data}} (A - A_m)^2 = [(A - A_m)^2] \tag{4-41}$$

the solution is given by

$$\frac{\partial M}{\partial z_0} = 0 \tag{4-42}$$

and

$$\frac{\partial M}{\partial Z} = 0 \quad , \tag{4-43}$$

where A is the computed value of the projected opening area (a function of Z and z_0), A_m is the measured value of the opening area, and z_0 is the z coordinate of the particle stopping point. Here again, as in the section on the tracing least-squares theory, the bracket notation is used to indicate a

summation over the data.

The computed track opening area projection, A, is obtained as follows. $A(\rho,\delta,\theta)$ is obtained from the subroutine $ETA(\rho,\delta,\theta,A,P)$ which generates A and the perimeter P from the values of ρ, δ, θ as per the theory described in the last section. The variables A and P are in the normalized units (units of length = $h_0/2$), so the incoming values of A_m are also normalized by dividing by $h_0^2/4$ prior to any computations. The value of theta is computed from the particle LET using the calibration equation

$$\sin\theta = \frac{1}{v_0 + \left(\frac{LET}{LET_c}\right)^{\psi}} , \qquad (4\text{-}44)$$

where v_0, LET_c, and ψ are the calibration parameters. They are obtained experimentally either from accelerator exposures to known particles or as an internal calibration from the particles in the stack identified to be iron. The variable LET is the restricted energy loss rate LET_{350}, the energy loss rate to electrons receiving energies less than 350 keV each. The units of LET and LET_c are keV/μm of H_2O. The LET versus range relationship is given by the logarithmic polynomial

$$\ell n\left(\frac{LET}{Z^2}\right) = \begin{cases} \sum_{n=0}^{6} d_n \left\{\ell n\left(\frac{Z^2R}{M}\right)\right\}^n & \ell n\left(\frac{Z^2R}{A}\right) < 9.093 \\ -2.192 & \ell n\left(\frac{Z^2R}{A}\right) \geq 9.093 \end{cases} \qquad (4\text{-}45)$$

With the units of LET in keV/μm of H_2O and R, the particle residual range, in mm of H_2O, the coefficients d_n are as given in Table 4.2.

Finally, the particle residual range is given by

$$R = \frac{|z - z_0|}{\sin\delta} , \qquad (4\text{-}46)$$

TABLE 4.2

n	d_n
0	1.1943
1	−0.47709
2	-1.314×10^{-3}
3	-1.4306×10^{-3}
4	1.20309×10^{-3}
5	-1.83602×10^{-4}
6	9.4199×10^{-6}

where z is the water equivalent height of the center of the layer in which the area measurement associated with z is made. Before the individual tracks are processed, a table of z values as a function of layer number is prepared by adding the water equivalent thicknesses of the layers, starting from the bottom of the stack, and subtracting ½ of the current layer thickness.

With the above set of equations, it is possible to compute A as a function of the layer number of the measurement and the parameters z_0 and Z. It can be seen, however, that because of the nonlinear nature of the dependence of A on z_0 and Z, Eqs. (4-42) and (4-43) cannot be directly solved for the values of z_0 and Z. Instead, it is necessary to linearize the expression of Z with respect to Z and z_0 by taking the terms through the 1st power in the Taylor series expansion. These linearized equations are then iteratively solved for Z and z_0 as will be described.

Before the iterative solution of the more rigorous equations can be started, it is necessary to find approximate starting values for z_0 and Z. To this end,

each of the measured areas is converted into an LET value by using Eq. (4-44) and subroutine ETA for computing the track area as a function of ρ, δ, and θ. Since area is thus specified as a function of LET rather than LET as a function of area, Newton's method in the form

$$LET_{i+1} = LET_i + \frac{A - A_i}{\frac{\partial A}{\partial LET_i}}$$

$$= LET_i + \frac{A - A_i}{\frac{\partial A}{\partial \theta_i}\frac{\partial \theta}{\partial LET_i}} \qquad (4\text{-}47)$$

is used to iteratively converge to the proper value of LET. The partial derivative of A with respect to θ, $\frac{\partial A}{\partial \theta}$, is approximated by

$$\frac{\partial A}{\partial \theta} = \frac{\Delta A}{\Delta \theta} \qquad (4\text{-}48)$$

where ΔA is the difference between the values of A from subroutine ETA from two nearly adjacent values of θ separated by $\Delta\theta$. From Eq. (4-44), $\frac{\partial \theta}{\partial LET}$ is given by

$$\frac{\partial \theta}{\partial LET} = \frac{\psi \tan \theta}{LET}(v_0 \sin \theta - 1) \qquad (4\text{-}49)$$

With the values of LET as a function of z now established, the power function approximation to the range-LET relationship,

$$\frac{LET}{Z^2} = c_L\left(\frac{Z^2 R}{M}\right)^{-\eta} \qquad (4\text{-}50)$$

can be converted into the form

$$LET^{-\frac{1}{\eta}} = \frac{Z^2}{M(c_L Z^2)^{\frac{1}{\eta}} \sin \delta}|z - z_0| \qquad (4\text{-}51)$$

Here R has been replaced by its value in Eq. (4-46). From Eq. (4-51), it can be seen that a linear regression fit of the form

$$LET^{-\frac{1}{\eta}} = f_1 + f_2 z \tag{4-52}$$

gives

$$z_0 = -\frac{f_1}{f_2} \tag{4-53}$$

and

$$Z = \left(M_1 c_L^{\frac{1}{\eta}} \sin \delta \, |f_2| \right)^{\frac{1}{2 - \xi - \frac{2}{\eta}}} \tag{4-54}$$

where M has been eliminated using Eq. (4-40). The linear regression is given by

$$\underline{f} = \underline{\underline{G}}^{-1} \cdot \underline{g} \tag{4-55}$$

where

$$\underline{\underline{G}} = \begin{pmatrix} [w] & [wz] \\ [wz] & [wz^2] \end{pmatrix}, \tag{4-56}$$

$$\underline{g} = \begin{pmatrix} [wLET^{-\frac{1}{\eta}}] \\ [wzLET^{-\frac{1}{\eta}}] \end{pmatrix} \tag{4-57}$$

and the weight, w, is given by

$$w = \left(\frac{\partial A}{\partial LET^{-\frac{1}{\eta}}} \right)^2 = \left(\frac{\partial A}{\partial \theta} \frac{\partial \theta}{\partial LET} \frac{LET^{\frac{1}{\eta} + 1}}{\eta} \right)^2 \tag{4-58}$$

The partial derivative $\frac{\partial A}{\partial \theta}$ is given by Eq. (4-48) and $\frac{\partial \theta}{\partial LET}$ is given by Eq. 4-49). The numerical values of the parameters c_L and η are 3.196 and 0.4247, respectively, for LET in keV/μm water and R in mm of water.

Having established the procedure for obtaining the starting values of Z and z_0, we now consider the iterative solution of the rigorous equations.

Equation (4-41) can be rewritten in the form

$$M = [(\Delta A - \Delta A_m)^2] \tag{4-59}$$

where

$$\Delta A_m = A_m - A_i \tag{4-60}$$

and

$$\Delta A = A - A_i = \frac{\partial A}{\partial z_{0_i}} \Delta z_0 + \frac{\partial A}{\partial Z_i} \Delta Z \tag{4-61}$$

is the difference between A and its approximation, A_i, given by the ith iteration values of z_0 and Z. The right-hand side of Eq. (4-61) shows the Taylor series expansion of ΔA in terms of z_0 and Z, where $\frac{\partial A}{\partial z_{0_i}}$ and $\frac{\partial A}{\partial Z_i}$ are evaluated using the ith iteration values of z_0 and Z. The deviations of z_0 and Z from their ith approximation values are

$$\Delta z_0 = z_0 - z_{0_i} \tag{4-62}$$

and

$$\Delta Z = Z - Z_i \tag{4-63}$$

respectively.

The partial deratives in Eq. (4-61) are given by

$$\frac{\partial A}{\partial z_0} = \frac{\partial A}{\partial \theta} \frac{\partial \theta}{\partial LET} \frac{\partial LET}{\partial z_0} \tag{4-64}$$

and

$$\frac{\partial A}{\partial Z} = \frac{\partial A}{\partial \theta} \frac{\partial \theta}{\partial LET} \frac{\partial LET}{\partial Z} \tag{4-65}$$

Equations (4-48) and (4-49) give $\frac{\partial A}{\partial \theta}$ and $\frac{\partial \theta}{\partial LET}$, respectively. From Eq. (4-45)

$$\frac{\partial \ \ell n\ LET}{\partial \ \ell n\ R} = \begin{cases} \sum_{n=1}^{6} n\, d_n \left\{ \ell n \left(\frac{Z^2 R}{M} \right) \right\}^n & \ell n \left(\frac{Z^2 R}{A} \right) < 9.093 \\ 0 & \ell n \left(\frac{Z^2 R}{A} \right) \geq 9.093 \end{cases} \tag{4-66}$$

and then from Eqs. (4-40), (4-46), (4-48), (4-49), and (4-64) through (4-66)

$$\frac{\partial A}{\partial z_0} = \frac{\psi(1 - v_0 \sin\theta) \tan\theta}{z - z_0} \frac{\partial A}{\partial \theta} \frac{\partial \ell n \text{ LET}}{\partial \ell n R} \tag{4-67}$$

and

$$\frac{\partial A}{\partial Z} = \frac{\psi(1 - v_0 \sin\theta) \tan\theta}{Z} \left\{(\xi - 2) \frac{\partial \ell n \text{ LET}}{\partial \ell n R} - 2\right\} \frac{\partial A}{\partial \theta} \tag{4-68}$$

With these expressions for the partial derivatives and the linearized expression for the sum of the residuals squared, Eqs. (4-59), (4-60) and (4-61) the least-squares solution is given by

$$\frac{\partial M}{\partial \Delta z_0} = \frac{\partial M}{\partial \Delta Z} = 0 \tag{4-69}$$

This leads to the solution for the correction to z_0 and Z,

$$\underline{a} = \begin{pmatrix} \Delta z_0 \\ \Delta Z \end{pmatrix} = \underline{\underline{H}}^{-1} \cdot \underline{b} \ , \tag{4-70}$$

where

$$\underline{\underline{H}} = \begin{pmatrix} \left[\left(\frac{\partial A}{\partial z_0}\right)^2\right] & \left[\left(\frac{\partial A}{\partial z_0}\right)\left(\frac{\partial A}{\partial Z}\right)\right] \\ \left[\left(\frac{\partial A}{\partial z_0}\right)\left(\frac{\partial A}{\partial Z}\right)\right] & \left[\left(\frac{\partial A}{\partial Z}\right)^2\right] \end{pmatrix} \tag{4-71}$$

and

$$\underline{b} = \begin{pmatrix} \left[\Delta A_m \frac{\partial A}{\partial z_0}\right] \\ \left[\Delta A_m \frac{\partial A}{\partial Z}\right] \end{pmatrix} \tag{4-72}$$

where the $\underline{a}$ and $\underline{b}$ and the matrix $\underline{\underline{H}}$ should not be confused with previous usage of such letters for other quantities. The RMS error in the measurements (from the dispersion about the curve) is given by

$$\sigma_A^2 = \frac{[\Delta A_m^2] - \underline{a} \cdot \underline{b}}{[1] - 2} \tag{4-73}$$

and the covariance matrix for the errors in Δz_0 and ΔZ,

$$\underline{\underline{C}} = \sigma_A^2 \underline{\underline{H}}^{-1} \quad , \tag{4-74}$$

is given by σ_A^2 and the inverse of matrix H. In Eq. (4-73), $[1] - 2$ is the number of measurements minus 2, the number of degrees of freedom of the system, and σ_A^2 is based on the assumption that chi squared = number of degrees of freedom. The quantity of great interest, the variance of Z, σ_Z^2, is given by the 2,2 component of the covariance matrix,

$$\sigma_Z^2 = C_{22} = \sigma_A^2 H_{22}^{-1} \tag{4-75}$$

As previously stated, the final solution of z_0 and Z is obtained by iteration. The i+1th iteration values are given by

$$z_{0_{i+1}} = z_{0_i} + \Delta z_{0_i} \tag{4-76}$$

and

$$Z_{i+1} = Z_i + \Delta Z_i \tag{4-77}$$

The iteration is stopped when the absolute value of Δz_0 and ΔZ are less than 0.01 mm of water and 0.01 atomic number units, respectively. The error analysis, Eqs. (4-73)-(4-75), is based on the results from the final iteration.

4.4 Calibration

In the previous sections it has been assumed that there are three calibration parameters, v_0, LET_c, and ψ, for each type of layer in the stack. It was also assumed that the value of these parameters is known from previous calibration experiments.

In practice, there is a fourth calibration parameter, the halation width, w, which describes the modification of the true experimental projected area to the actual measured area. This difference is produced by optical halation at the periphery of the etched-through hole. The halation phenomenon is depicted in Fig. 4.5 where the true hole opening, the apparent hole opening

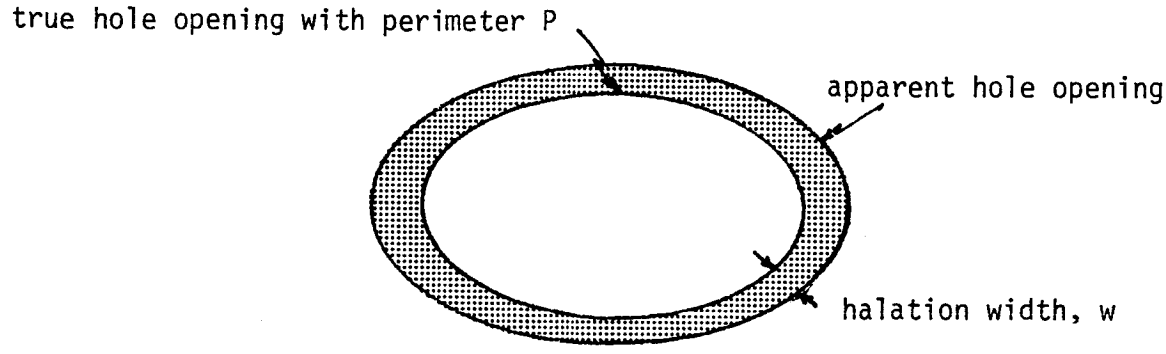


Figure 4.5. Halation correction. The apparent hole is larger than the actual hole by the halation width, w, given by the shaded halation correction shown.

and the halation correction are shown for an elliptical hole. In the figure w is positive but may also be negative depending on the precise detection level for the sample. From the figure, it can be seen that a reasonably good estimate of the area correction is the perimeter of the hole, P, times w. Thus, the actual computed area used in the computations is

$$A_c = A + Pw \quad , \tag{4-78}$$

where A is the area computed from the etch-through geometry described in Section 4.2. For actual experimental data, it is A_c rather than A that is used in the evaluation of Z described in Section 4.3. This also means that the partial derivative $\frac{\partial A}{\partial \theta}$ becomes

$$\frac{\partial A_c}{\partial \theta} \simeq \frac{\Delta A_c}{\Delta \theta} = \frac{\Delta A}{\Delta \theta} + \frac{\Delta P}{\Delta \theta} w \qquad (4\text{-}79)$$

With a constant halation width assumption, the rigorous expression for A_c, $A_c = A + Pw + \pi w^2$, could be used rather than Eq. (4-78). This expression has the practical limitation that there is a discontinuity in A_c at A and P just equal to zero because, for larger values of θ, $A_c = 0$ and, for slightly smaller values of θ, $A_c \simeq \pi w^2$. This mathematical discontinuity creates a serious obstacle to the convergence of the iteration for z_0 and Z. In the real physical case, w is variable as the hole approaches the diffraction-limited size and the measured area goes smoothly to zero. Therefore, the approximation of Eq. (4-78) is used rather than the rigorous expression based on the assumption of constant w.

For the actual measurements made on a detector stack, the values of the calibration parameters v_0, LET_c, ψ, and w as determined from a previous calibration may not be perfectly applicable. This is because of unavoidable differences in the sample exposure environment, the storage time and conditions, the processing conditions, and the detection level during the digitizing process. What is usually possible to achieve and certainly is strived for is that all of the layers of a given type within a given stack be handled in nearly the same manner as possible. This helps to insure internal consistency in the measurements. To evaluate the small changes in the calibration parameters, it is necessary to use the population of particles within the stack.

The internal calibration procedure can be outlined as follows. 1) The approximate calibration previously determined is used to reduce the data. 2) The Z spectrum is plotted for the reduced data and the peak due to some predominant particle species, such as iron, is identified, even though the peak may not lie precisely at the expected value of Z. 3) The measured particles which fall within the defined bounds of Z from Step 2 and which stop in

the stack are selected for the calibration procedure. They are all assumed to be the nominal particle and each stopping point is assumed to lie in the center of the first layer either not having a track or having a subsize track at the high-LET end of the particle's set of measurements along its trajectory. 4) The calibration parameters are adjusted to produce the proper statistical properties of the identified particles. Typically, the halation width is adjusted to move the computed stopping point as nearly as possible to its nominal position in the center of the next layer and LET_c is adjusted to correct the average value of the computed Z. It is rarely necessary to adjust v_0 and ψ.

4.5 Statistical Analysis

After each of the measured particles has been analyzed to establish its trajectory, stopping point and identity, statistical quantities based on these parameters or functions of these parameters are derived. The most general form that these statistics can take is distributions of the parameters and their functions which are pertinent to dosimetric evaluations. Examples of such distributions are the LET spectrum and the particle Z spectrum.

From the general distributions, the more compact statistics such as dose, dose equivalent, and fractional cell loss can be computed. Of course, the computation of the directly biologically oriented statistics, dose equivalent and fraction cell loss, rely on having the results of biological experiments which relate the biological effectiveness of the radiation to the independent variables of these distributions. The virtue of having the results of the HZE analysis in the more general distribution form is the fact that these biological relationships are not all well known and are subject to change. Therefore, this final analysis step (the reduction to dose equivalent and fractional cell loss) is not considered to be part of the automated HZE process. The physical quantity, dose, can be readily computed, however biologically significant it

may or may not be. It should also be noted that the restriction to HZE particles places certain qualifications on the reduced statistics, such as dose and dose equivalent, because only particles with LET above a cutoff value for particles with $Z \geq 6$ are registered. Thus it is necessary to compute such quantities as being Z and LET specific, for example, the dose due to particles with LET > L_0 and $Z \geq 6$. In the general dosimetry packages, which include detectors other than stacks of plastics, such apparent deficiencies are obviated by the further measurements made in the other detectors. These measurements sample those portions of the spectrum not covered by the HZE stack.

Experimentally estimated distributions usually take one of two forms, differential or integral distributions. To obtain the experimental estimates of such distributions, the measured events are often binned according to the measured parameters of the events. Then the number of events collected in each bin is divided by the geometric factor for the bin. The geometric factor is essentially the bin size used in collecting a number of events for an experimental measurement of a frequency distribution. Another way of defining the geometrical factor is to say that it is the n-dimensional area of the domain of acceptance in parameter space. In this case the parameters are those which identify the event. In the strictest sense, these parameters should be geometrically related parameters such as positions and angles for the "geometrical" factor to be labeled as such. We would like to extend the definition to include other parameters such as time, energy, atomic number, etc., for the many common situations in which distributions of nonspatially oriented parameters are considered. A few familiar examples will serve to illustrate the concept of the geometrical factor.

Assume that we want to determine the fluence of ions from an accelerator. We place a detector in the beam at normal incidence and irradiate. If the particles are high-LET particles at normal incidence, they will etch with 100%

efficiency. We then count the number of particles in a given area and divide by the area over which the particles were counted. This gives the fluence or the number of particles/area = dN/dA. In this case, the area is the geometrical factor. The particle parameters are the x and y coordinates of their positions on the detector. The parameter space is just the x,y plane and the acceptance domain might be $x_1 < x < x_2$ and $y_1 < y < y_2$. The area is, of course, $A = (x_2 - x_1)(y_2 - y_1)$. Thus, the fluence could also be considered the distribution in x and y since $dN/dA = d^2N/dxdy$. The most usual approach to counting the tracks is to define the area to be counted by some visual boundary such as lines on the eyepiece reticule. Only those tracks within the boundary are counted. An alternate approach would be to measure the position of all the tracks within and slightly beyond the intended boundaries. The actual selection of the tracks would then be accomplished mathematically by seeing if the acceptance criteria $x_1 < x < x_2$ and $y_1 < y < y_2$ were met. This is the more usual situation for some of the parameters involved in cosmic-ray measurements because we cannot delineate the particle energy, for example, by a line on a reticule.

In the same experiment, we might want to introduce an additional parameter in which the events are distributed. If we had measured the time of the exposure, we could divide the fluence by this time to obtain the flux or the number of particles per (area × time). In this case, the generalized geometrical factor is the area times the time and its units are "distance^2·time".

Usually, when we are measuring distributions, the distribution is a function of some or all of the parameters in which the events are distributed. To measure this dependence either of two approaches is followed: 1) a differential distribution is obtained by breaking the domain in parameter space into several sections. This may be done by dividing the axes corresponding to any number of the parameters into sections. These sub-domains are then typically

called bins. The number of particles falling within the bin is divided by the bin size (the geometrical factor) to obtain the value of the distribution at the values of the parameters somewhere within the bin (usually at its center). Often the differential distributions which vary with only one parameter are displayed as histograms. 2) An integral distribution is obtained by ordering the events according to one or more of the parameters. Let's assume one parameter, which we will call the independent variable. In this case, the geometrical factor is the area in a parameter space consisting of only the other parameters of the distribution, not the independent variable. If the geometrical factor is not a function of the independent variable, the integral distribution is then just the number of events with the value of the independent variable greater than some reference value (the abscissa of the plot) divided by the geometrical factor due to the other parameters. If the geometrical factor is a function of the independent variable, it is computed for each event and when the sum is formed, the reciprocal of the geometric factor is added for each event with a value greater than the reference value.

As an example of differential and integral distributions, consider the measurement of the particle stopping density as a function of Z. Let's assume for simplicity that we have a stack of volume V, and that we can detect and identify every particle stopping in the stack. The differential distribution is then the number of particles/volume-Z = $\frac{d^2N}{dVdZ}$. Assume that the number of particles only warrants using bins of width $\Delta Z = 5$. We might then chose bin boundaries at 2.5, 7.5, 12.5, 17.5, 22.5, and 27.5 in Z. Then the number in each bin divided by 5V is the value of $\frac{d^2N}{dVdZ}$. The bin for $22.5 < Z < 27.5$ would then establish $\frac{d^2N}{dVdZ}$ at Z = 25, etc. The integral distribution for the reference value Z_0 would simply be $\frac{dN}{dV}(Z > Z_0) = \frac{N(Z > Z_0)}{V}$.

If the volume V is a function of Z, however, the variable geometric factor approach must be taken. Let's assume V = V(Z). Such a situation would oc-

cur, for example, if not only the particle stopping point but also a Z dependent length of the particle trajectory were required to be within the stack. Then the differential distribution in the bin $Z_1 < Z < Z_2$ is either

$$\frac{d^2N}{dVdZ} = \frac{N(Z_1 < Z < Z_2)}{(Z_2 - Z_1)\, V\left(\frac{Z_1 + Z_2}{2}\right)}$$

or if greater rigor is desired

$$\frac{d^2N}{dVdZ} = \frac{1}{(Z_2 - Z_1)} \sum_{Z_1<Z<Z_2} \frac{1}{V(Z)}$$

The integral distribution is given by

$$\frac{dN}{dV}(Z > Z_0) = \sum_{Z>Z_0} \frac{1}{V(Z)}.$$

Notice that in the integral distribution the differential, dZ, is not present.

4.5.1 LET Spectrum

In this section we examine the specific application of the principles just outlined in determining an LET spectrum from the measured data. If we could idealize the situation and assume that the solid-state track detectors have only an LET cutoff and that this LET cutoff is not dip angle dependent, we could then assume that all tracks with LET above this cutoff would register, whatever their dip angle might be. In this case, the projection factor solid-angle product is just a constant. Since the average projection factor for an isotropic beam is ½, then, if we consider that particles are traveling in both directions, that is, into the detector and out of the detector, the total solid-angle projected area product for the detector area in question would be 2π times this area. Then the LET spectrum could be assembled by just considering the measured LET values for particles with LETs above the cutoff for the detector and treating them as was outlined in the general discussion of geometric factors.

In the real case, solid-state nuclear track detectors have a dip-angle

cutoff which is LET dependent. This is because the track cone angle is given by the arc sine of the bulk etch rate of the plastic divided by the track etch rate of the plastic. Since the track etch rate is LET dependent, this means that for high-LET tracks with large track etch rates the cone angles are small. The detector will register all tracks with dip angles greater than the cone-angle value in the idealistic case where we detect them under a microscope. With a scanner, which is the case that we are dealing with, we are searching for etched-through holes. Therefore, there is another practical cutoff which is still LET dependent but is slightly different than the above-mentioned criterion. However, this cutoff can be defined mathematically. The cutoff that exists in the detector then implies that only particles with dip angles greater than the cutoff values will register. This means that, to obtain the effective solid-angle projection product, we must integrate above this cutoff value of dip angle instead of integrating the dip angle from zero to $\frac{\pi}{2}$. Therefore, we find that the geometric factor, which in this case is solid-angle projected-area product, is LET dependent. Its general nature is that for low values of LET just slightly above the absolute cutoff for the material, only a small solid-angle window is available and ultimately for stopping heavy particles the solid-angle projection factor almost approaches the maximum of 2π times the area of the detector.

The actual LET spectrum is produced by considering the tracks intersecting a small-thickness stack. This stack consists of several layer types to cover all values of LET. For each layer type and with consideration of the processing and readout given the layer, an effective LET cutoff is determined. The effective LET is the LET cutoff of an ideal detector with an absolute dip-angle-independent cutoff which would give the same measured track fluence as the real detector when placed in an isotropic field of particles with an LET spectrum having an assumed approximate slope. The measured fluence on each of

the detector types is considered to be the value of the integral LET spectrum at its effective LET cutoff. With detectors of several different LET cutoffs, it is possible to obtain several points along the integral LET spectrum curve. If the slope (on a log-log plot) of the spectrum significantly differs from the slope assumed in the effective LET cutoff determination, the new slope can be used to slightly change and correct the effective LET cutoff. With the improved cutoffs, the spectrum is redetermined and the process is continued to convergence, which requires very few iterations.

4.5.2 Z-Energy Spectrum

Deriving the Z-energy spectrum is a much more involved process than the process of deriving the LET spectrum. This is because of the complicated geometry of any real stack. If we imagine a rectangular stack and a particle impinging upon this stack at arbitrary dip and azimuthal angles and at an arbitrary energy, we must ask the question, "What is the projected area that the stack presents to this particle over which it can be stopped in the stack and register?". In asking such a question, it is already presumed that collection criteria for the particles have been established. In this case, the implicit assumption is that the particle is required to stop in the stack. To illustrate this concept, consider particles normally incident upon the stack. The minimum detectible energy is the energy which is sufficient to cause a particle to register in at least three layers of the stack, since this is one of registration criteria. For any energy between this minimum energy and the energy necessary for the particle to just pass through and stop in the final layer of the exit side of the stack, the projected area seen by this stack is the lateral area of the stack. If we consider particles deviating somewhat from normal incidence, slightly more area would be seen because it is possible for a particle to enter from the side of the stack and yet stop in the stack.

However, the full energy particle previously mentioned would be essentially limited to the same area. If we consider higher-energy particles, that is, those with higher energy than that necessary to just traverse the stack, the particles cannot enter at normal incidence and still stop in the stack. They must enter at an inclined angle so that they can go obliquely through the stack to provide more path length in the stack. For such an inclined track, the projected area of the stack is somewhat reduced over the normal incidence condition. Moreover, at a given energy, we see that we are also limited in solid angle. As we pass to the very highest energy detectable by the stack, we see that a particle of this energy traverses the stack from one corner on the top to the opposite corner on the bottom of the stack. Obviously, for such a particle, the projected area is extremely small as well as the range of possible dip and azimuthal angles. Thus, the solid-angle projection-factor window for particles with energies slightly less than the maximum possible energy is very small and passes smoothly to zero as the energy goes to the maximum energy. With the above considerations about the projected area as a function of the space angle, it is possible to obtain the solid-angle projected-area factor by integrating over the solid angle, that is, over all the possible values of the space angles. The integral is the integral of the projected area and the domain of integration is the range of all possible space angles for which there is a finite projected area for the particle and energy in question.

In the outline of the energy-spectrum computation in the previous paragraph, the registration criteria were, first, that some small finite thickness was necessary to register the track and, second, that the stopping point of the particle must lie within the stack. Otherwise there were no restrictions on the particle. In practice, several other restrictions need to be placed on

the particle. Not only must the particle register in three layers for it to be identified by the tracing program but it must actually register in several more layers to obtain an accurate Z value. This distance may even be Z dependent. In addition, there may be overlying material which serves as an energy degrader. All these aspects of the problem must be taken into consideration in the final analysis of the geometric factor for the Z and energy-spectrum reduction.

5. HARDWARE

5.1 General Automated System Versus Specific Minimal System

In describing the hardware used in the automated HZE analysis, it is necessary to distinguish between the general automated system and the specific minimal system. The general automated system consists of all the hardware used to do the many and varied tasks which can be accomplished in the area of automated HZE. This includes many other automated measurements other than the measurement of etched-through holes at relatively low magnification. Several other automated measurements have been made in addition to the specific HZE readout as dictated by this contract. The equipment to be described in the following section includes not only the equipment purchased under the present contract but other equipment. It is being described because it all relates to the automated HZE facility at the University of San Francisco and has some relevance in the area of automated track measurement. In contrast, the specific minimal system is the minimal hardware necessary for the performance of automated HZE as outlined by the contract. The minimal system consists of a vidicon connected to an optical digitizer which is then interfaced to the PDP-11/34 computer. For the actual performance of automated HZE, it is only necessary to have the basic 32 K of core memory. In addition for the minimal system, it is necessary to have a stepping stage to scan the samples and computer peripherals such as a terminal, two RK05 disk drives, and a Versatec line printer/plotter for storage, input, and output of the data. These pieces of hardware will be described with the rest of the system in order.

5.2 PDP-11

Our current computing facility consists of two PDP-11 series computers. One of these is a PDP-11/10 which is interfaced to a Quantimet image analysis system. At the present, this is essentially a dedicated computer with rela-

tively few peripherals other than the Quantimet. In addition, the 11/10 is relatively less powerful than our other central processer, therefore the majority of the data processing is accomplished on our other processer.

Our second 11 series processer is a PDP-11/34. It has a full complement of 128 K of MOS memory, a DZ-11 multiplexor connecting to up to 16 terminals when operated under an appropriate operating system, two DL-11 interfaces for serial line data, a floating point processer to enhance the processing speed, and an RK05J disk drive with a non-removable double-density platter and a removable single-density platter. Together these platters make three logical RK disk-drive units, each capable of storing 1.2 megawords of data. Also interfaced to the computer is a TU-10 mag tape unit and two RK07 28 megabyte disk drives for mass storage of data. Currently our complement of terminals consists of a teletype unit, the Eyecom digitizer terminal which also serves as a standard terminal, a Tektronix 4012 video terminal with a storage screen capable of making online plots, a DEC VT-52 refresh memory video terminal and two Soroc video terminals. Hard copies are produced on a Versatic 200 point per inch printer/plotter.

5.3 Quantimet

Our original digitizer was the Quantimet image-analyzing computer. This is a hardware system consisting of a vidicon interfaced to hardware which displays a video image, detects this image by grey-level, and analyzes for features. In addition, the Quantimet is capable of other functions but these are not necessarily directly related to the automated HZE application. The Quantimet system consists of a number of modules. A list of the modules for our system is a 1-D detector, an MS-3 standard computer, two function computers, a classifier-collector unit, a variable frame and scale unit, a feature image field interface which is used to interface the output of the Quantimet to the computer,

and a control interface which allows the computer to control the various functions of the Quantimet. The Quantimet has a resolution of 880 points in the horizontal direction by 688 points in the vertical direction. These points are scanned with a nonstandard scan time of 1/10th of a second. The scan is a non-interlaced scan. As a field is scanned, the points are processed in real time to first detect the points by grey-level threshold. Then they are assembled into features in the standard computer. These features are then discriminated as to size and other possible parameters after they have been transmitted to the computer via the FIFI interface.

Although the Quantimet has large resolution and is reasonably fast in the hardware mode, as a digitizer it is very slow. It will digitize but it takes 88 seconds per field to feed out the individual grey-level information for all of the 600,000+ pixels in the field. Another disadvantage of the Quantimet is its poor performance and service record. The Quantimet is also quite expensive. Our current investment is well over $100,000. For these reasons, the ultimate digitizing for automated HZE will not be performed by the Quantimet. Instead, A straight digitizer interfaced to the computer has been chosen. This approach will ultimately allow a more thorough analysis of the features because the individual gray-level information from each pixel is available.

5.4 Eyecom Digitizer

The Eyecom digitizer is a standard TV scanning system with both a high-speed and a high-resolution image digitizer. For the purpose of automated HZE the high-speed digitizer is used because this increases the readout speed. The field is a standard 525 line TV format field and it is scanned in an interlaced fashioned in 1/30th of a second. The high-speed digitizer uses only the first half of this interlaced scan and digitizes the full frame in 1/60th of a second at a resolution of 320 points wide by 240 points high. The grey-

level information is accurate to 5 bits or 1 part in 32. This information is stored in the refresh memory of the Eyecom system where it is available on a pixel by pixel basis to the computer. The Eyecom system resides directly on the PDP-11/34 bus and its control, status, and data registers are accessed in the same manner as any other memory location on the 11/34 bus.

5.5 Microscopes

The basic approach to automated HZE does not require the use of a microscope since the plastics are digitized using noncompound magnification at approximately 1 to 1 scale between the plastic and the vidicon face. Microscopes are used for special purposes, however. One of these purposes is to look at tracks in detail. This mode will be accomplished by first digitizing the plastic at low magnification without a microscope and locating the tracks. Then the stage of the microscope will be driven to the location of the tracks as measured at low power and they are digitized in detail. Obviously, this is a very slow process in the case of high track fluences but is warrented if high accuracy is required. For very low fluences, the multiple scan approach should be more efficient because the tracks are quickly found at a very low power which is inadequate to produce sufficiently accurate measurement. Essentially this approach eliminates the need for scanning the large area between tracks at high power.

The microscope we use for automated track measurements is a custom-built microscope, originally designed for emulsion measurements, which has been outfitted with a stepping stage and a stepping z motion. The Quantimet or the Eyecom cameras may be placed on this microscope to view the field as it exists under the microscope. The stepping stage is interfaced to either computer via the DR-11C interface and standard stepping motor drivers.

A second automated microscope is used for semi-automatic measurements. In this case, the digitized stage can be driven under manual or computer control. It has encoders so that when it is driven under manual control the computer can sense its location. An operator uses the microscope and control panel to transmit locations of track features and/or orientations of tracks via a DR-11C interface.

5.6 Stepping Stages

In addition to the stepping stage on the semi-automated microscope just described, we have two stepping stages that are useful for automated HZE. The microscope stepping stage has increments of two microns in x and one micron in y. The stage has a four by four cm travel and is driven by standard stepping motors which advance one of the increments per pulse generated. It is interfaced to the computer via a DR-11C and another simple hardware interface which allows manual control of the stage as well. The schematic for the electronics that allow manual control is shown in Fig. 5.1. The output of this electronics goes to a standard stepping-motor driver manufactured by Slosyn.

A second stepping stage is the one actually used with the Eyecom for the low-resolution digitizing. Its step size is 10 microns in both x and y. This means that it can travel at high rates of speed, that is, up to one cm per second once the stage has accelerated to its maximum speed. This stage electrically is identical to the first stage being driven via a DR-11C interface interfaced with the same electronics shown in Fig. 5.1. The stepping stages described are driven from software controls, each pulse being generated by the computer. The advantage of such an approach is that constant acceleration ramping can be employed to achieve the maximum ramping rate.

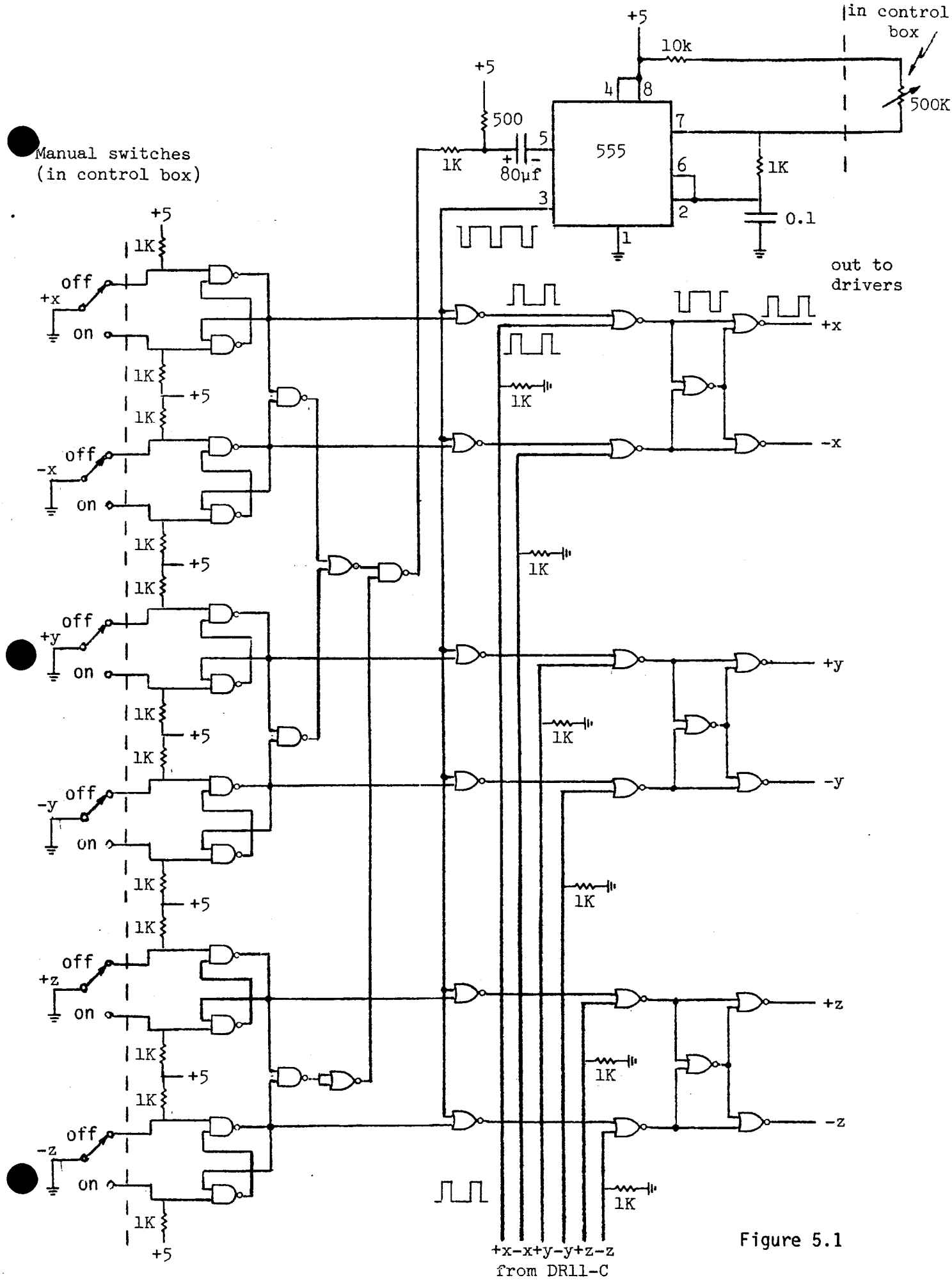

Manual switches
(in control box)
+5
10k
in control
box
500K
500
555
1K
80µf
0.1
out to
drivers
off
on
+x
-x
+y
-y
+z
-z
+x-x+y-y+z-z
from DR11-C

Figure 5.1

6. SOFTWARE

The automated HZE software will be described at two different levels. The principal programs will be described in fairly great detail. These descriptions will illuminate the overall approach to the solution of the problems posed at each of the steps in the data reduction procedure and will also deal with some of the specific details of the programs. The details dealt with are those sections of code that may seem superfluous at first glance but which were incorporated for a specific reason to enable the correct execution of the task at hand.

In addition to the principal programs is a miscellaneous group of generally small programs for display, printout, parameter file generation, plotting, testing, etc. These programs will be priefly described in alphabetical order in Section 6.6.

Listings of all the programs related to the automated HZE work will be given in alphabetical order in the appendix. They are given in the form employed under the RT-11 operating system. Most of the frequently used programs, other than the digitizing program, have also been converted to operate under the RSX-11M operating system. The description of the changes required for this conversion will be given elsewhere.

6.1 Parametric Analysis for Stack and Processing Design

The parametric analysis consists of two programs. The first of these programs calculates the efficiency for registering a given particle type as a function of the layer configuration employed in the stack. This configuration can include up to three materials. This program, labelled AUEFF3, calculates the fraction of the particles which stop in the stack which will have at least three measurements per particle. This is the registration criterion employed because any fewer measurements do not allow enough redundancy to

confirm the existence of the track. AUZER3 assumes an isotropic distribution of particles.

The second parametric analysis program is labelled AUZER3 and is an analysis of the error in Z expected for the given configuration being tested. Again this program allows three different layer materials and, of course, different processing regimes and layer thicknesses as well. It also includes the effect of having a restricted stack thickness, that is, a thickness such that the heavy particles can register through the entire stack. In this case, the number of measurements per particle is limited by the stack thickness rather than the sensitivity of the detectors. AUEFF3 calculates the uncertainty expected as a function of Z and allows a proper balance between insensitive and sensitive detector layers. This is necessary because, if sensitive detector layers are used, high-Z particles saturate near the stopping point and cannot therefore produce accurately measurable Zs unless a very thick stack is employed. Even if a thick stack is employed, the problem of fragmentation occurs. Therefore, it is desirable to have, in addition to the sensitive material, insensitive material which will give track etch rate measurements which are not saturated near the stopping point for heavier particles. The sensitive material is necessary for lighter particles.

The theory of the AUZER3 program is quite similar to the theory used in the analysis of the error in Z in the Z computation program, AUZC. It differs, however, in that the range-LET relationship is a simplified power-function relationship, similar to the one used to obtain the first approximation in the Z analysis.

6.2 Data Taking

The objective of the main-line track measurement program is to provide position and area measurements for all of the tracks on each of the sequential layers in the stack. These layers will be measured one material at a

time. The first step in the measurement process is to establish the grey-level correction for the Eyecom scanner field. This is necessary because the illumination is not perfectly uniform. This is accomplished by looking at a blank or defocused field of the sample which has a minimal number of tracks in it and also looking at a clear field with the detector layer removed which is comparable to a track area which has been etched through. An arbitrary fraction of the distance between these two grey levels for each point is then the grey-level threshold. The grey levels actually used in processing the data are stored in the main memory of the computer. Therefore, the averages of several adjacent pixels are used to establish a relatively coarse matrix of grey-level cutoffs over the Eyecom field. This consists of approximately three-hundred areas over the Eyecom field.

After having established the grey-level cutoff array, the detector layers are digitized one at a time. The detector layer must be scanned in a raster pattern because the track size is very small compared to the overall size of the detector layers. Therefore, to obtain the desired resolution of tracks, it is necessary to use a number of relatively small Eyecom fields to cover the detector area. Prior to the scanning of each detector layer, its individual coordinate system is established by the computer which initially places the position origin fiducial in the alleged center of the field. The operator then places a cursor on the actual origin and its position within this field is fed to the computer. The computer then advances to the approximate position of the second fiducial, which is also measured by the operator with the joy-stick cursor. These two measured positions on the sample then define its precise coordinate system relative to the coordinate system of the scanning stage and the Eyecom. As the tracks are subsequently measured, their coordinates are translated and rotated to reduce them to the standard reference system of the stack.

The track measurements within a given Eyecom field are made using the

digitized array of grey-level values from the Eyecom. A 240 by 320 pixel array is digitized into the refresh memory of the Eyecom in 1/60 of a second at a resolution of 5 bits or 1 part in 32 grey levels. This information is read out by the computer after it has translated the stage to a new position. The reason for the translation preceding the data readout is that the new image can be stabilizing on the Vidicon photosensitive surface while the data is being processed by the computer.

The data is processed column by column finding the detected areas of features based on the grey-level threshold. Within each column, the pixels are examined to find those which are above the grey-level threshold cutoff. The features are light against the dark background because of the light transmitted through the etched-through holes. This initial step, which is the most time-consuming aspect of the pattern recognition task because it involves every pixel in every field, is written in DEC assembly language to provide the maximum speed possible. The intersection of each vertical column with the feature is then output as one piece of data to a higher-level FORTRAN subroutine. This FORTRAN subroutine assembles features from the individual columns by looking for contiguous sections of a given track. Then the position and area of this track relative to the Eyecom field is transmitted to the main-line program. The main-line program converts these positions into absolute positions and the areas into absolute areas and the data is stored on the output disk files.

After all of the layers of a given material type have been digitized, the same process is repeated for each of the other material types. Then the data from the individual material types is combined so the information from sequential layers comes in order in the final output file.

6.3 Tracing

The theory of the tracing process has already been described. This theory

is implemented in a program called AUTRAJ. The program accomplishes two ends. First, it reorganizes the data. As previously described, the data originally is obtained in order of layers, that is, for each layer all the track positions and areas are read out and then subsequent layers come in order. What is desired, however, is that the information for each particle rather than for each layer be contiguous in the file. A second role played by the tracing program is to analyze the trajectories of the particles, that is, to establish the azimuthal and dip angles as well as the intersection with the reference layer which has been chosen to be the level in the stack at which z is equal to zero, which is usually the bottom of the stack. The program also establishes the uncertainty in the trajectories, specifically the uncertainty in the dip angle of the particle. The deviation of the measurements about the fit trajectory are output as well. This information allows the spurious trajectories, that is, those resulting from misidentification, to be discerned because the residuals in such cases are usually too large. Previously, it was mentioned that at least three track measurements are necessary for each particle. One of the reasons for having three is that this allows such discrimination to take place. If two measurements alone are employed, a trajectory will fit perfectly, even if these are not in reality associated.

The mechanics of the tracing program are as follows. Several files are used to store both input and output information. One disk file contains all of the calibration information for the various layers in the stack. Each layer is allowed to have a unique thickness and fraction of the layer removed as well as a unique set of calibration parameters. For the tracing program, only the layer thickness is important. The data from the input calibration file is output to a temporary file which has the information about the layer thickness and height, that is, the distance from the origin in z for each layer. Two additional temporary files are established to contain the incomplete trajecto-

ries. This information is passed back and forth between these files, once for each layer encountered.

The initial step in processing the data once the parameter file has been established is to read the data from the first layer. Each track measurement on this layer becomes a trajectory in the temporary trajectory file #1. Then the following layers are read in sequence. On each layer, the data is read into the computer memory and ordered by increasing x values. This facilitates comparison of the established trajectory with the data. Once the data from the layer has been read into the memory, the established trajectories on the previously written temporary trajectory file are compared with the incoming data. For each match, the data is incorporated with the trajectory and a least-squares fit is made to establish the temporary values of the trajectory parameters. This information is then written to the output temporary trajectory file and the input data is flagged since it is only to be used with one trajectory. After all the trajectories have been compared with the input data, two steps are performed. The first is to establish a new trajectory for every track from the input file which has not been matched with the previously established trajectory. Also the previously established trajectories are searched for those which have been completed, i.e. those which have not encountered any track measurements for more than a certain number of layers, called the layer gap. The reason for allowing gaps in the trajectories is that occasionally layers are less sensitive. For instance, if a sequence of layers is being used, it is desirable to be able to track across the insensitive layers to the next sensitive layer. It is necessary, however, to require that trajectories terminate after a certain layer gap, otherwise the number of trajectories in the temporary file would increase almost without limit. Moreover, spurious identifications would be introduced because the program recognizes the uncertainty in the intersection of an established trajectory with the layer currently being examined. As one proceeds farther and farther from the

section of trajectory which has already been encountered, this position on the layer becomes more and more uncertain, hence, more and more nonsignificant track measurements could be incorporated into this trajectory.

Finally, when the whole stack has been searched for measurements, all the remaining temporary trajectories are output to the output file as completed trajectories. This completes the operation of AUTRAJ for a single set of track measurements.

6.4 Z Analysis

Following the determination of particle trajectories, the data in the file output by AUTRAJ is analyzed by the Z-analysis program, called AUZC. AUZC receives the data from each particle individually. The input data consists of the trajectory parameters as well as the measured values of the track areas on each of the layers. It initially establishes a temporary paramater file which contains now the height, not in physical dimensions such as mm but in water equivalent range. This allows detectors of mixed types to be conveniently used in the stacks. For the plastic materials used, which are very similar to water in their stopping characteristics, it is quite accurate to just scale both the range and the energy loss rate within the material to that of water.

As the individual trajectories are processed, several steps are accomplished. The first of these is to establish, in addition to the area measurement for each particle, an LET value. By comparing the area with the area calculated using the known calibration parameters from the parameter file and adjusting the LET until a match is achieved, it is possible to establish the LET at each layer level. Then it is possible to eliminate the erroneous end measurement due to a stopping track because as one proceeds along the trajectory toward the stopping point the LETs increase until we reach the end of the track. For the layer in which the particle actually stops, however, an erroneous measurement is often produced because, although it etches through, the tra-

jectory has not passed through the entire layer and an etched-through area which is too small is produced. This leads to a value of LET which is too small. So, in principle, by looking for the maximum value of LET and seeing if this occurs one layer from the end, AUZC can eliminate the measurement from the stopping layer. In practice, the logic of this step is somewhat more complicated because of the fact that the track cone angles go essentially to zero near the particle stopping point. The actual logic can be seen in the listing of the program in the appendix.

The next step in the process is to establish the initial value of Z and the particle stopping point, which will later be refined in an iterative process to obtain the exact values. This step assumes a power-function range-energy relationship. From this relationship it follows that LET raised to the appropriate constant power is linearly related to the range of the particle. Then a linear regression to this parameter as a function of water height in the stack gives the intercept and slope which are translated into the particle stopping point and Z, respectively, as described in the theory section.

The final step in the Z analysis for a particle is to establish the accurate values of Z and stopping point. These are established by using an iterative linearized least-squares fit. The reason it is necessary to use a linearized least-squares fit is that the calculated track area is not linearly related to the undetermined parameters. As already described in the section on theoretical background, this nonlinear relationship is made linear by using the first term in the Taylor expansion of the area measurement with respect to the parameters. Thus it becomes a standard linear least-squares fit. After the first application, a correction to the approximate Z and stopping point is achieved and subsequent steps improve these approximations to the point of convergence.

After the final converged solution has been achieved, the various data are output to another file. In this file is contained the original trajectory information which has been passed on from the input file as well as the information obtained from the Z analysis. The data output by the Z analysis program is the computed value of Z and its uncertainty, the computed value of the stopping point and its uncertainty, the correlation of these two uncertainties, and the number of iterations necessary for convergence. Finally, the measured and the computed values of the track areas as normalized to one-half the layer thickness are output. The output file is then subsequently analyzed to find statistical relationships between the particles both in Z and in stopping point.

Although the Z computation is in principle quite straightforward, a number of small features were required in the program to insure convergence and the proper fitting of the data. Each of these features will be discussed at this point but not necessarily in the order in which they occur in the program.

The basic track model assumes that the projected areas can be computed from the known fraction of the layer removed, the original layer thickness, and the geometry of the track, specifically the dip and cone angles. The actual measurements, however, deviate somewhat from this theoretical area in that a small band around the theoretical area is either included or excluded from the measurements, depending on the grey-level sensitivity setting. This phenomenon is known as halation and is a quite familiar property of optical measurements. A correction for this phenomenon consists of computing the measurements not by the theoretical area alone but by also including a halation width times the theoretical perimeter of the feature. The halation width in this sense is one of the calibration parameters that must be determined to fit the data. Without the halation width correction it is found that the computed stopping points deviate from their proper values.

If a track-area measurement is larger than the theoretically maximum area possible, no LET can be derived for this track. Therefore, in the initial approximation, the reciprocal of LET is set to zero. However, in the final iterative convergence, these measurements are included and the residuals for them always remain positive. This then enforces the stopping point being very close to such a measurement because it implies the track is near the stopping point hence the cone angle is very small.

A feature used twice in the program is the fractional approach to convergence, that is, the corrections to the parameters being sought are computed and only a fraction of these is added to the previous approximation. This approach is used both in the original computation of LET values prior to the initial Z and stopping-point approximation and also in the final iterative convergence. The purpose for using such a fractional convergence approach is that if the full correction were added for the initial steps of an iteration, often the solution is "over shot". This may lead to an oscillation or divergence rather than convergence. As the iterations continue, the fraction of the correction that is added in each case is increased and approaches the full correction as convergence is very nearly approached. In some cases, when it is found that the solution has somewhat diverged, again the fraction is reduced to help toward achieving the final convergence.

In all cases, it is required that the stopping point of the particle lie outside the range of measurements. This physically means that no negative residual ranges are allowed. Therefore, if on one step in the iterative process the correction to the stopping point would lead to a computed stopping point lying within the range of the measurements, the actual stopping point established at this step is forced to remain outside this range but is allowed to asymptotically approach the last measurement point.

If the measurements indicate an approximately constant LET value throughout the measurement set, then the program would normally compute a stopping point which is very far away. If this distance exceeds a certain maximum value, the implication is that the tracks are suprarelativistic, that is, the particle energy at this point is allegedly higher than the energy at which the relativistic minimum in ionization occurs. This is an unrealistic situation in practice because the measurements of stopping point and Z therefore become meaningless. In such cases, then, the stopping point is not determined and it is assumed that the track is relativistic and that the Z is determined on this basis. Such results are generally not applicable to a real analysis in that they would produce systematically different results because a different category of track is involved. In reality, the particle is probably not completely relativistic. This is ultimately not a problem, however, in that such measurements are flagged by giving them a zero correlation coefficient. Thus, in the statistical analyses which follow the Z-analysis program, such events are discarded unless a stack known to actually contain relativistic particles, such as one exposed during a low-latitude mission, is being processed.

6.6 Statistical Analysis

Although of interest for analyzing the performance of the program, individual values of Z and stopping point are of little meaning in an evaluation of the biological significance of the radiation field in the spacecraft. To obtain information pertinent to biological assessment, it is necessary to assemble the results output by the Z-computation program into statistical quantities, that is, distributions which can then be integrated over with various weighting factors to determine the biological significance of the results. This step then is to establish such statistical parameters.

The details of the statistical analysis depends not only on the layer configuration within the stack but also on the overall configuration of the stack involved. This is because the energy of incidence, for example, for a particle is dependent not only on where it stops but how much material it passed through in coming to that point. In addition, other information such as even the shielding configuration of the spacecraft could be taken into account in arriving at a spectrum which existed outside the spacecraft. Therefore, the statistical portion of the data analysis is a very customized step, only applicable to a specific configuration. Accordingly, the discussion in this section is quite general, serving as a guideline to the generation of future programs applicable to specific exposure situations that will be encountered.

The LET spectrum can be determined with a relatively thin stack. The currently employed method of determining the LET spectrum was described in the theory section. It can be employed with stacks of as few as two or three layers. Another method which requires a somewhat thicker stack would use the automated HZE-tracing and Z-analysis programs.

The basic approach is to use just enough layers to establish the particle trajectories with reasonable certainty. In addition, within these layers are materials which represent different sensitivities so that different portions of the LET spectrum can be obtained. The process consists of measuring these layers in the same manner as a large stack would be measured. Likewise they are traced and Z analyzed in the same way that a normal stack would be analyzed. The difference between such a thin stack and a thick stack is that the values of Z and the stopping point established will be essentially meaningless if used as such because only a short section of the registration range of the particle is contained within the stack. However, these quite uncertain Z and stopping-point measurements combine to produce an accurate LET value at the level of the measurements within the stack because this value of LET was

the essential result measured in the stack. The quantity that is missing to allow accurate Z and stopping-point measurements is the rate of change of LET across the stack. But, of course, in the LET spectrum analysis, this is not the interesting quantity. Therefore, because of the high correlation between the error in Z and the stopping point, it is possible to obtain the LET value at some reference level within the stack. It is preferable to have this level close to one of the least sensitive layers in the stack because it is for the particles which have the high-LET values measurable in the low-sensitivity layers that the LET is most rapidly variable.

The actual analysis of the LET spectrum is as follows. A single thin stack is used to establish the LET spectrum. Its greatest sensitivity lies in the direction normal to its surface. As particles with trajectory angles deviating more and more from this surface-normal direction are encountered, the effective sensitivity of the stack decreases because the area projected in the direction of the beam decreases and because there is a dip-angle cutoff which is dependent on LET. That is, all particles with a given LET will register at dip angles above some critical value. This means that for a given value of LET, a certain solid-angle window is available to the detector. As the LET increases, this window becomes larger. The LET spectrum that is to be determined basically is the number of particles per projected area per solid angle per LET interval. In this form the spectrum is the differential LET spectrum. A more practical form for the experimental spectrum to take is the integral LET spectrum, that is, the number of particles per projected area per solid angle interval with LET greater than the cutoff value of LET. In this form, the statistics for all particles with LETs greater than the cutoff accumulate and the curve obtained is a smoother one. Then this experimental integral LET spectrum can be fit by some analytic expression which can then be differentiated to obtain the differential LET spectrum.

The integral LET spectrum is produced by first ordering each of the measurements in LET. For each of the LET values, the corresponding geometrical factor, that is, the projected area solid-angle window, is computed. Then with the integral LET spectrum initialized to zero at large values of LET, the reciprocal of the appropriate window is added to the LET spectrum as each particle is encountered in the ordered array of decreasing LETs. This process is described in mathematical form in the section on theoretical foundation.

The Z spectrum obtained from a thick stack or orthogonal configuration of thick stacks is obtained in a similar manner, that is, the particles are collected into bins according to Z and some other parameter which may be stopping point or maybe energy. For each of these bins, the window in solid angle is obtained and the integral spectrum is derived by summing from the largest Zs to the smallest Zs. For each particle encountered, the reciprocal of the window is added to the integral spectrum. This process is also described in mathematical terms in the section on the theoretical foundation.

When both the LET spectrum and the Z spectrum are obtained, a tacit assumption was made that there is a cutoff in dip angle or some other parameter that establishes the window. This corresponds roughly to the experimental cutoff but is chosen to be somewhat more restrictive than the actual experimental cutoff. Then in the statistical analysis process, the more rigorous cutoff is mathematically applied to the actual data. This means that some of the data obtained is discarded because it is near the region of marginal registration in the phase space. In other words, only the portion of the phase space for which nearly 100% registration can be assumed is selected. Particles outside of this region of phase space are discarded so that the effective experimental cutoff is identical to the one used in the theoretical computation of the window.

6.6 Miscellaneous Programs for Display, Printout, Parameter File Generation, etc.

Program	Function
ASUM	A FORTRAN-callable assembly routine to form the sum of the values in a 16 by 16 array at an arbitrary location within the Eyecom refresh memory. It is called by the FORTRAN routine SHADE in the process of determining the grey-level cutoff array.
AUALS	Accomplishes the selection of the output of AUZC by the azimuthal angle. Thus, all the particles in a given azimuthal angle range can be selected.
AUCALP	Used to plot the data from a calibration layer. It is assumed that this layer is a layer generated by a high-energy, heavy-particle exposure using a wedge energy degrader which produces an array of tracks on the layer with linearly varying particle residual range along the layer. Such a layer would be digitized as any other layer in an automated HZE stack but the data from the layer has the above characteristic, namely, that the energy of the particle and the residual range of the particle varies along one axis.
AUCL	Used for manually cleaning up the measured data after such measurements have been collected by the tracing program. This program plots the points on the screen of the Tektronix and when the cursor is placed above a point and an "N" entered at the terminal, the particular point is removed from the data that is to be processed by the Z-analysis program.
AUCNT	Used for counting the number of particles in an AUZC output file.
AUCOMB	Combines an arbitrary number of AUZC output files into one composite file.

AUCTOL Used for converting the calibration layer measurements into a form amenable to least-squares solution, that is, in the file the data is plotted in the order of area as a function of position along the strip.

AUDIPS Used to select the output of AUZC by dip angle. The data is then output to a file of similar form but with the data selected in a given dip-angle range.

AUDIRS Used for selecting the output of AUZC by the direction of travel of the particle. Again the program is output to a similar file.

AUDISS Used for selecting the output of AUZC by the distance of the particle stopping point and again the data is output to an output file of a similar form.

AUHIGH Plots out the height in the stack as a function of layer both in mm and in mm of water for the purpose of correlating data on various layers with the height given in the output files given in AUTRAJ and AUZC.

AUITH Used for storing the data in a form that can produce histograms of the number of iterations.

AUNH Used for writing the number of tracks per particle from an AUTRAJ output file into a histogram file.

The four following programs generate, modify, and print the data in parameter files. The parameter files are the files used as input information on the individual layers in the stack. They are unique for each stack in that they contain the information about the configuration within the stack.

AUPAR Generates the parameter file with the assumption that all layers of a given material type are identical. However, within this constraint, an arbitrary configuration of these layers is possible.

AUPARC Allows changing of certain elements of a parameter file.

AUPARI Allows for individual variation from layer to layer even within a given material type. The assumption for this program is that all the layers may be of slightly different thickness.

AUPARP Used to output the parameters of the individual layers. It produces a table containing all of the calibration parameters for all of the layers in a stack.

AURCOM Used for combining raw data files from separate sections of the stack into one file.

AUREGP Allows the registration range in CN, Lexan, and CR-39 to be plotted as a function of the dip angle with Z as a parameter.

AURLVX Used for plotting raw data points. In this plot, the abscissa is the layer number, the ordinate is the x value of the point on the layer and the angle of the line used to designate the point represents the y value on the layer.

AURMER Used to merge separate raw data files into a composite file. The data from the input files is merged in the proper layer sequence. This program is typically used to combine the measurements made in the different types of detector layers that have been interspersed in a composite stack.

AURPR Used for printing the raw data. Because of the volume of raw data, this program is of limited usefulness because a very large printout is required to demonstrate all of the raw data.

AURTLP Similar to AURLVX except that, in this case, the abscissa is the x value, the ordinate is the y value and the angle of the line is the layer number indication.

AURTPP Again similar except that a point is used for each raw data point. In this case, the abscissa is x and the ordinate is y. The reason

for this plot, in contrast with the previous ones, is that smaller points are possible and a higher track density can be visualized.

AURTTP Combines the raw data which is printed in the same form as the program AURTLP plots it and the output of the program AUTRAJ which, as has been described, generated the program trajectories. This is a comparison program to analyze the performance of AUTRAJ.

AURVLP Used for plotting the raw data output by AUSTO on the Versatec printer plotter. Each of the points measured is plotted in its proper position. All of the data from the whole stack is superimposed on one plot, however, the layer which was read is indicated by the orientation of the line used to plot the point.

AURVPP Writes the track positions contained in a raw data file to an x,y file.

AUSAH Used for generating histogram files for the sigma of the area, namely, the residual about the fit curve output by the program AUZC.

AUSDH Used for producing a histogram file for the uncertainty in the determined dip-angle values. It operates on AUZC output files.

AUSEL Used for selecting the traced trajectories with more than a minimum number of tracks per trajectory. It is used primarily for testing purposes in which case only events having a large number of tracks per trajectory are used.

AUSELA Used for selecting particular trajectories. In this case, individual trajectories with given values of azimuthal and dip angle are selected. This program again is used for testing purposes after particular tracks with problems of convergence or some other difficulties have been selected and need to be used as test tracks for rectifying problems.

AUSH Used to make a histogram file of the spread of the measured points about a trajectory. It operates on the output from AUTRAJ.

AUSTOE Takes data from a low-power scan for use as a map to indicate the location of each of the tracks for subsequent measurement by program AUSTOF.

AUSTOF A data taking program used following the use of AUSTOE. It measures each track individually at higher power rather than scanning the layer.

AUSVZ Plots the stopping point calculated for a particle versus its atomic number. This program operates on files output by AUZC.

AUSZH Produces a histogram file for the uncertainty in Z values output by program AUZC.

AUTADP Generates an x,y file for the α and δ values output by AUTRAJ. The purpose of the x,y file is to then make a plot of x,y points.

AUTCOM Combines AUTRAJ output files.

AUTDH Generates a histogram file for the dip angle values output by AUTRAJ.

AUTERH Generates a histogram file for the RMS deviation of trajectories as output by AUTRAJ.

AUTNH Generates a histogram file for the number of tracks per trajectory as output by AUTRAJ.

AUTNHT Generates a table of the number of trajectories with n layers per trajectory

AUTPL Plots the output of AUTRAJ. In this case, what is plotted is the track area as a function of layer number for individual particle trajectories. In addition, the parameters of the trajectory and their uncertainties are printed out at the top of the plot.

AUTPR prints the parameters of trajectories output by AUTRAJ.

AUXYPL Sets up x,y points for plotting on the Versatec. The input to this must be ordered in x.

AUZCG Plots the output of the Z-analysis program AUZC, the measured, as compared to the computed, values of the track areas. The measured values are given as points; the computed values are given as a line on the plot. In addition, the parameters generated for the particle are printed out at the top of the screen.

AUZCOM Combines output files from AUZC.

AUZCP Prints the output of AUZC.

AUZCPS An abbreviated output of the parameters generated by AUZC. It includes only the particle parameters and not the individual track parameters as are included in AUZCP.

AUZH Generates a histogram file for the output of AUZC.

AUZNH Generates a histogram file for the number of measurements per particle as output by AUZC.

BPOINT A subprogram to plot a point on the screen of the Tektronix terminal.

COLUMN A FORTRAN-callable assembly routine which analyzes one column of the Eyecom refresh memory to find segments of the column which are composed of points which are above the grey-level threshold.

DIG A FORTRAN-callable assembly subprogram used to digitize the image from the Vidicon into the refresh memory of the Eyecom.

ETA A subprogram used to compute the normalized etched-through area and perimeter of tracks as a function of the fraction of the layer removed and the particle dip and cone angles.

ETAPLT plots the etched-through area as computed by ETA as a function of the track cone angle, theta.

ETAPRT Prints the etched-through area and perimeter as computed by ETA.

ETAVRP Plots the etched-through area as a function of the particle resid-

ual range for particles with a simplified etch rate calibration relationship.

ETPPLT Plots the etched-through perimeter as a function of the cone angle.

FEATUR A subprogram which assembles the detected line segments output by COLUMN into features composed of contiguous detected points in the plane. The number of features, their coordinates within the Eyecom field, and their areas are returned to the calling program.

FX A subprogram called by the least-squares-fit routine, LSF. In this form FX with LSF accomplishes a least-squares fit of a polynomial to the input x,y data.

GHIST Plots a histogram which is composed of a sum of gaussian distributions. The standard deviation of these gaussians is arbitrary and each is centered at a data point. This contrasts with the standard histogram which is a plot of "binned" data values.

GRAPLT Plots an arbitrary array of dots on the Versatec printer/plotter. The input data comes from an unformatted sequential file. Each bit of each input record determines if one of the points on the corresponding line is to be plotted. Originally designed to make grey-level plots, in the context of automated HZE data processing, GRAPLT is the final program in the sequence of programs AURVPP, XYSORT, AUXYPL, GRAPLT used to plot the raw data directly on the Versatec as very small dots which are consistent with high track fluences.

HAIRS A subprogram which brings up the Tektronix cross-hair cursor and then returns the x,y position of the cross-hairs to the calling program as well as the ASCII representation of the keyboard character used to initiate the coordinate transfer.

HIST Plots a conventional histogram.

IHIST Plots the integral distribution of the values from an input file.

LET A subprogram which places the Tektronix alphanumeric cursor at a specified location on the screen prior to the printing of a character.

LSF Produces and plots a least-squares fit of a linear combination of arbitrary functions of x to a set of measured x,y points contained in an x,y file. The arbitrary functions of x are computed by the subroutine FX.

LSFCAL A modification of LSF used to determine calibration parameters from the output of AUCTOL.

LSFCL Allows the interactive removal of erroneous data from an x,y file which is to be the input file for LSF.

MATINV A subprogram which is a standard library matrix inversion routine.

MTX An assembly subprogram which is a proprietary routine from Versatec for outputting plotting information to the Versatec printer/plotter. No listing of MTX is included in the appendix.

POPL Produces point plots on the Tektronix terminal from x,y data contained in an x,y file.

SCALE A linear-scaling and axis-drawing subprogram for the Tektronix terminal.

SURFA A subprogram which replaces ETA when the measured feature is the surface opening of a track rather than the etched-through hole.

TRAN A subprogram used to cause a specified translation of the macro- or microscope stage. Included in the same file are the translation initiation routine TRANI and another routine called by TRAN, TR.

VECTOR A subprogram which plots a vector on the Tektronix terminal.

XYSORT Orders the records in an x,y file by x using a merge sort procedure

The programs described above (with the exception of MTS) are given in the appendix as listings. The main programs are fairly well documented. Many of the supplementary small programs were only intended for testing purposes and for that reason have not necessarily been well documented but should be understandable because they are usually somewhat brief.

7. SYSTEM PERFORMANCE

7.1 Parametric Analysis Results

The parametric analysis program has been run for a large number of possible layer and processing combinations. Of those examined, the optimum processing combinations have been reduced to those using the maximum possible etch on each layer consistent with having a final thickness greater than 40 microns. This lower limit of thickness is governed by the requirement that secondary particles, such as those induced by protons, not be etched through. Since such secondaries have ranges generally less than 40 microns, restricting the final thickness to being greater than 40 microns essentially eliminates all of these secondaries. This then means that for the three materials currently used, Lexan with a thickness of 0.19 mm is etched to remove 0.8 of the layer, CN with a thickness of 0.1 mm is etched to remove 0.6 of a layer, and CR-39 with a thickness of 1 mm is etched to remove 0.9 of a layer. The final thickness of the CR-39 was chosen greater than 40 microns because of the variability of the CR-39 layer thicknesses so the thinnest layers will actually approach the 40 micron figure whereas the thicker layers will not reach this figure for the one etch given the whole stack.

To compare the individual materials and to see the effect of combinations, the parametric analysis program computes two quantities of interest, σ_Z, which is the uncertainty in Z expected for the particular configuration involved and the efficiency of detection. For the efficiency of detection program, it is assumed that the beam is isotropic. In practice this may not be a good assumption but for the parametric analysis purpose this is quite adequate. Also, it is assumed that the particles encounter a stack thickness of at least 1 cm for the Z analysis. It is important to specify this dimension because the accuracy in Z is quite dependent on this thickness for the more sensitive materials since particles are required to stop in the stack. This situa-

tion arises because, for sensitive materials, the final portion of the trajectory near the stopping point can be saturated in track area. Therefore, essentially no information is available except that the particle is higher than a certain value in Z. If a greater range of trajectory is available, then the falloff in the etch rate at these greater residual ranges is sufficient to calculate the Z. This minimum thickness has been chosen as 1 cm of water. This means that, for a several-cm-thick stack, a majority of the particles stopping within the stack will be acceptable for a Z analysis.

The results for individual material types are shown in Figs. 7.1, 7.2 and 7.3. These are the results for Lexan, CN and CR-39, respectively. For each of these, σ_Z and efficiency is given in parts a and b. It can be seen that the Lexan is both inadequate in Z resolution and efficiency for low-Z particles but performs well for high-Z particles. The opposite is true for CN. CR-39 is somewhere between the CN and Lexan in its performance. Although the CR-39 is actually more sensitive in the sense that, for normal incidence tracks, lower LETs can be registered than in CN, its performance is considerably different in detail than that of the CN. Specifically, CN saturates (the cone angle goes to zero) at high-LET values much lower than the LET value of saturation for CR-39. That is, the etch rate versus the LET curve is much flatter for CR-39 than it is for CN. Therefore, the CN will have large etched-through holes for a fairly large portion of the trajectory and only at the greater residual ranges will the etch rate and hence the hole size decrease. For CR-39, the converse is true. Only near the stopping point will the holes be full size; then very quickly the holes decrease to a fairly small size and they remain small out to very large values of the residual range. Thus, the CN and the CR-39 are complementary materials, and CN and CR-39 taken as a pair are in turn complementary to Lexan.

The performance of combination stacks are shown in Figs. 7.4a,b and 7.5a,b. Figure 7.4 shows the performance of a stack of interleaved Lexan and CN layers.

Figure 7.1. a) The expected error in Z and b) the efficiency of a stack of 190-μm-thick Lexan polycarbonate layers. The calculation assumes that the digitization pixels are 20 μm on a side and that 80% of the layers have been removed during the etch. In addition, the analysis of the error assumes a typical dip angle of 52 deg and that the measurements are contained within a vertical distance of 1 cm from the particle stopping point. The efficiency is defined as the fraction of the particles stopping within the stack which produce at least three measurable tracks within the stack. The efficiency calculation assumes an isotropic beam of particles.

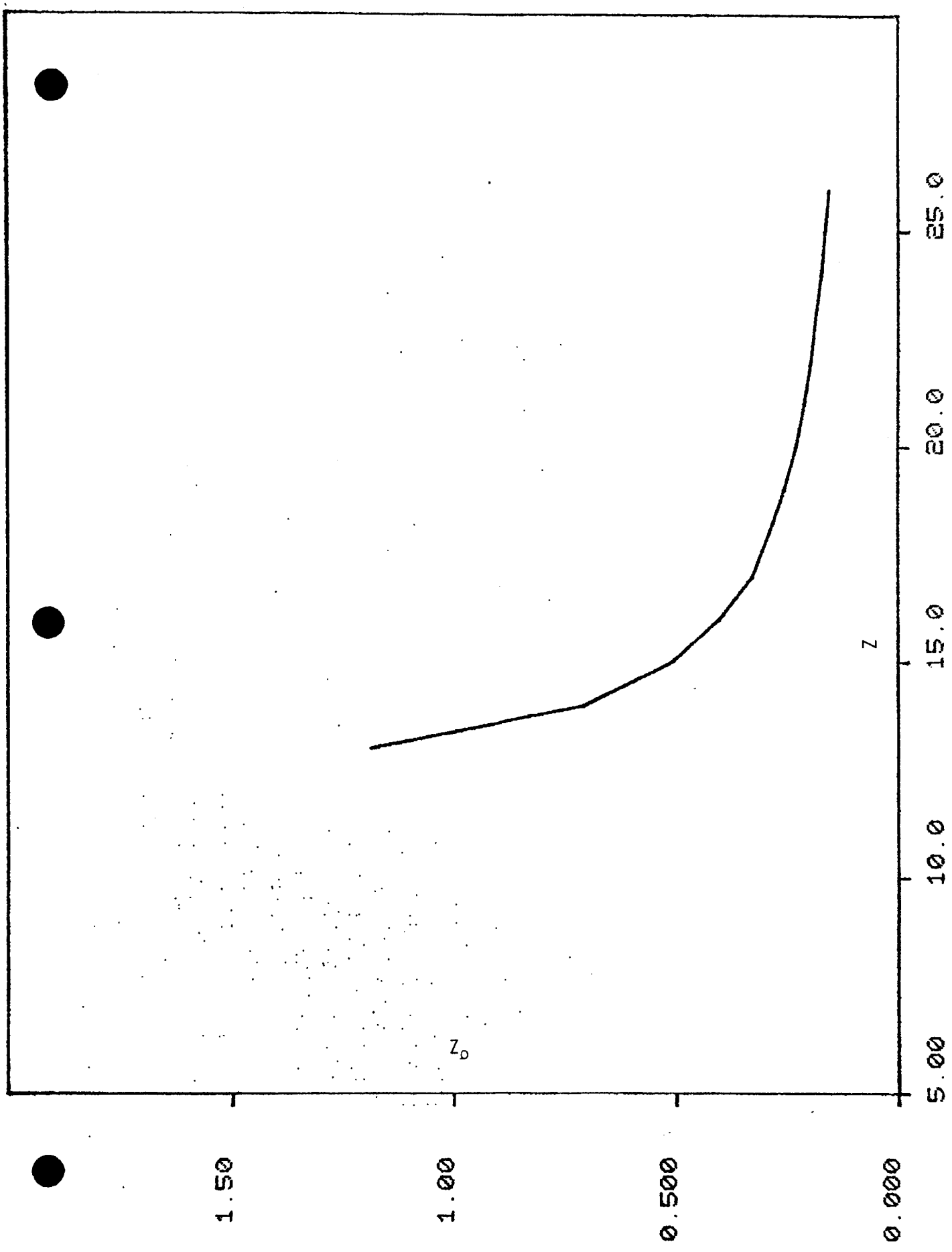


Figure 7.1a

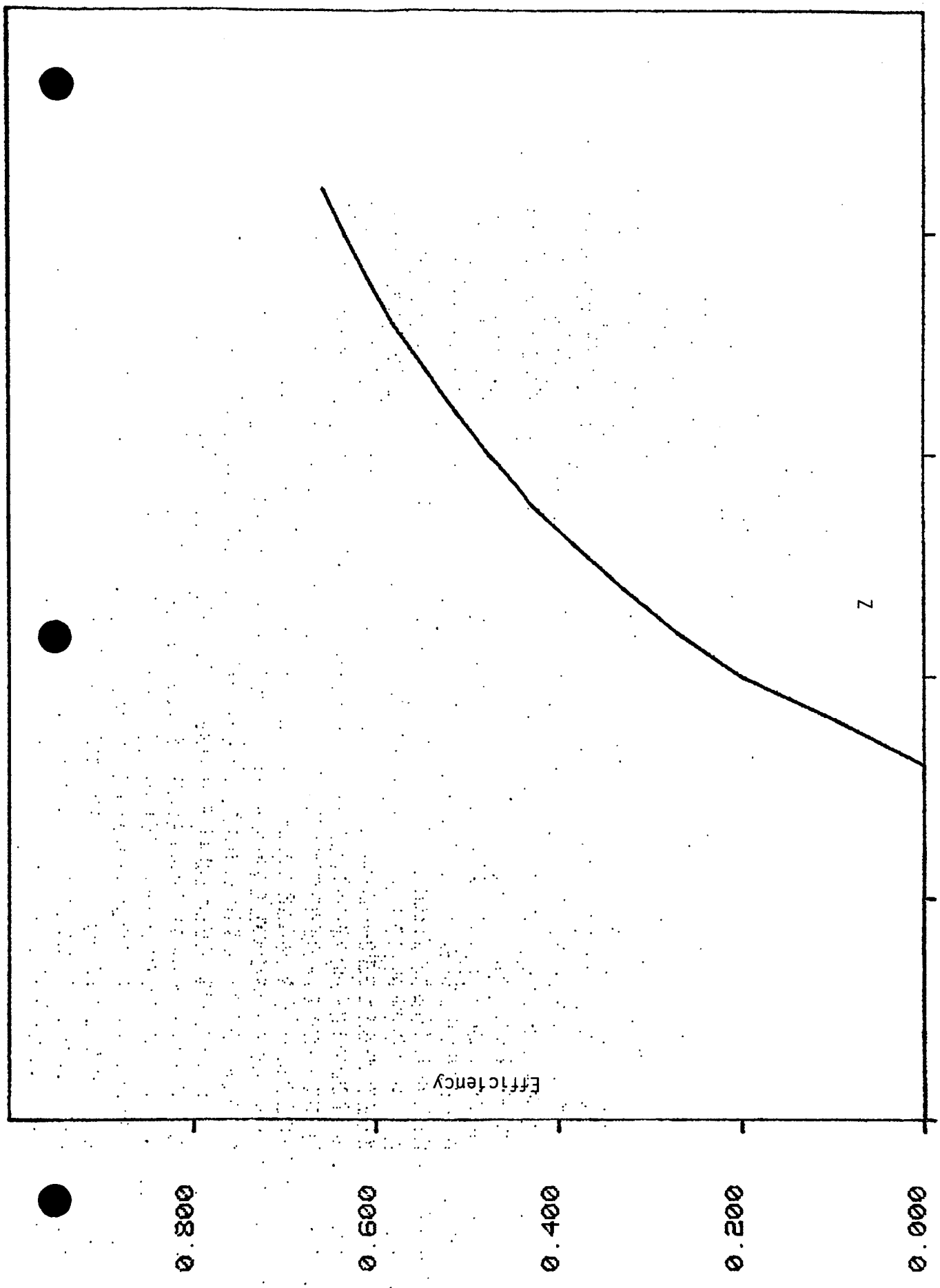


Figure 7.1b

Figure 7.2. a) The expected error in Z and b) the efficiency of a stack of 100-μm-thick cellulose nitrate layers with 60% of the detector layer removed during etch. The other assumed conditions are the same as for Fig. 7.1.

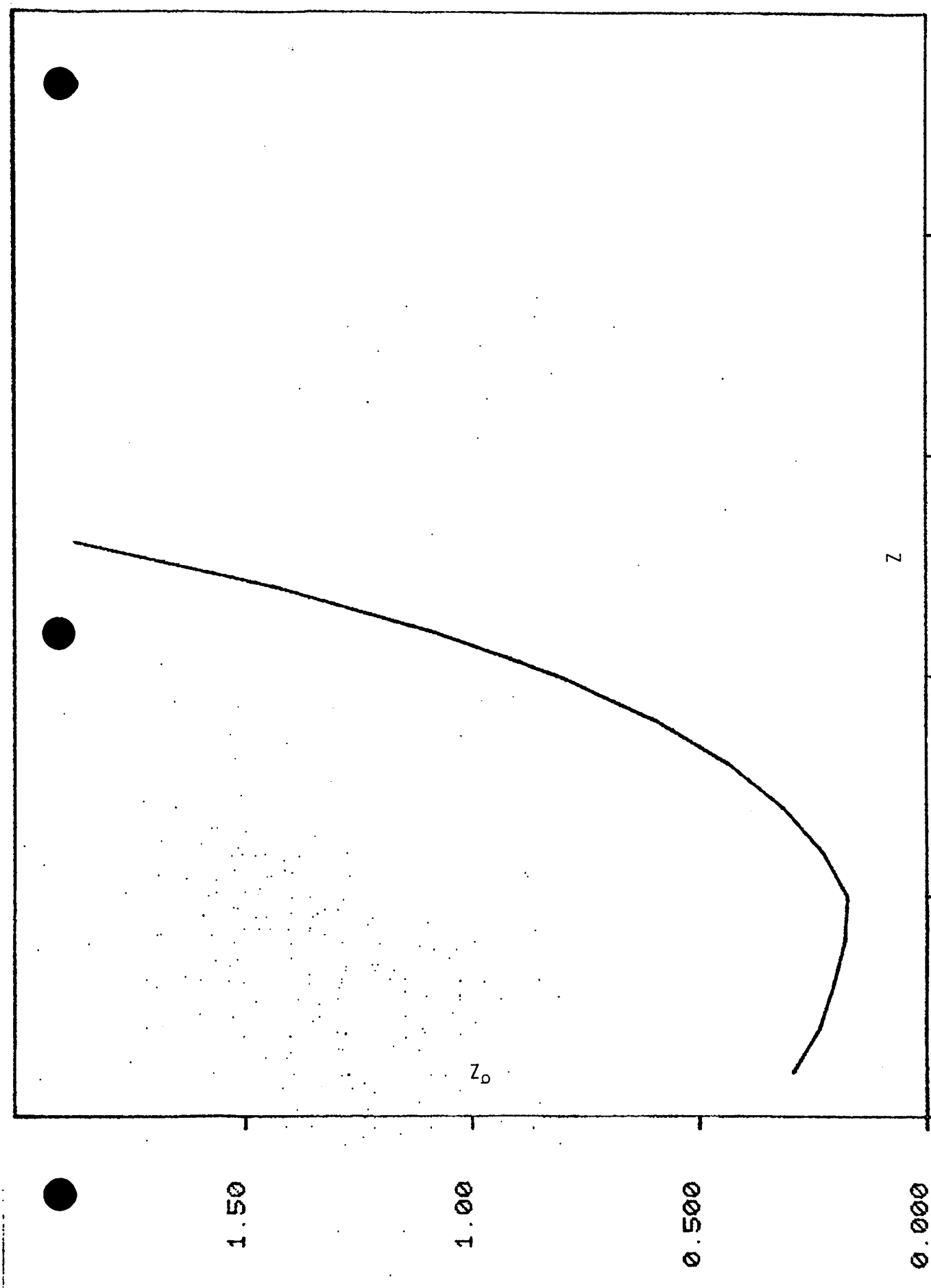


Figure 7.2a

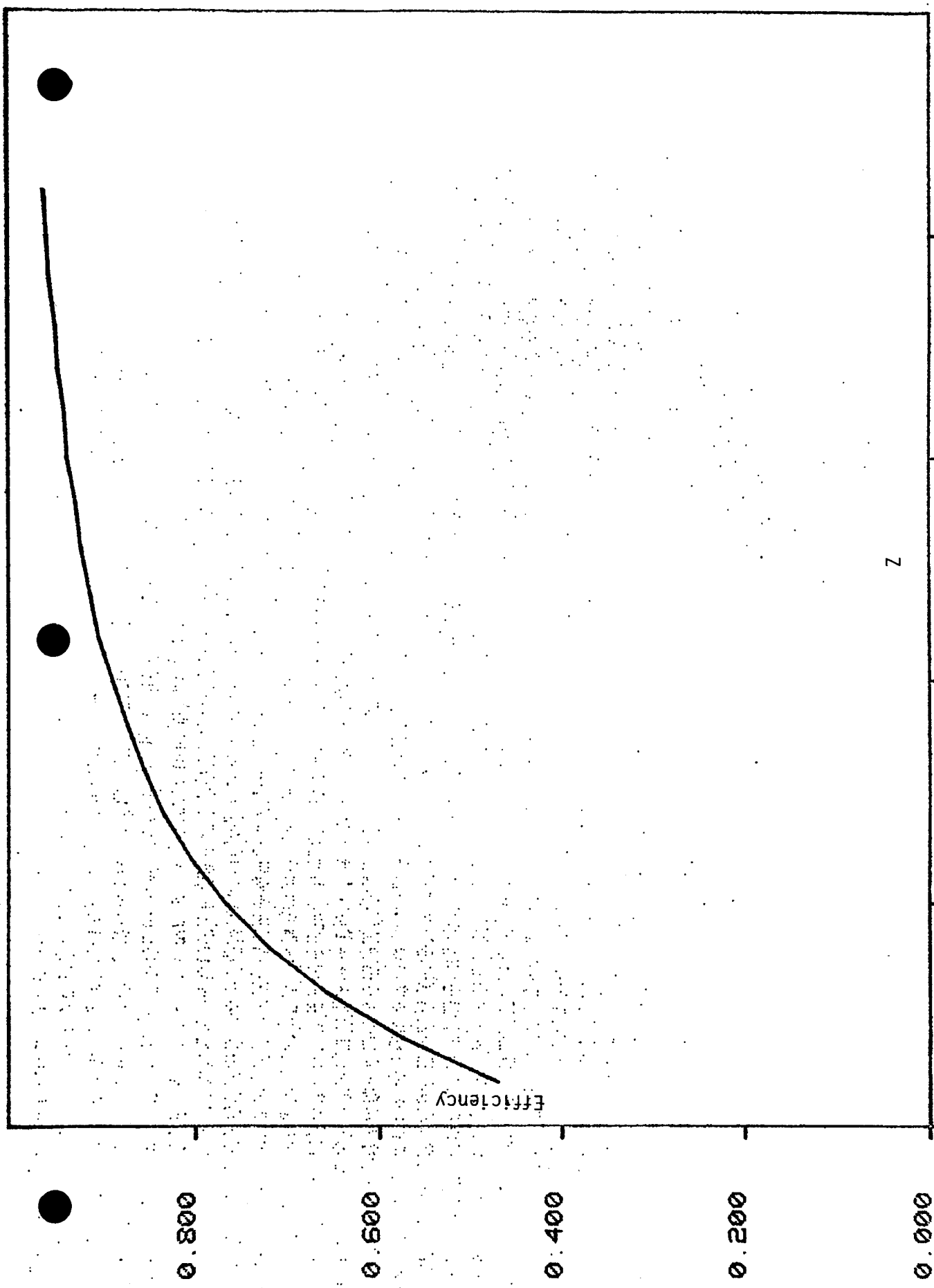


Figure 7.2b

Figure 7.3. a) The expected error in Z and b) the efficiency of a stack of 1-mm-thick CR-39 layers with 90% of the detector layer removed during etch. The other assumed conditions are the same as for Fig. 7.1.

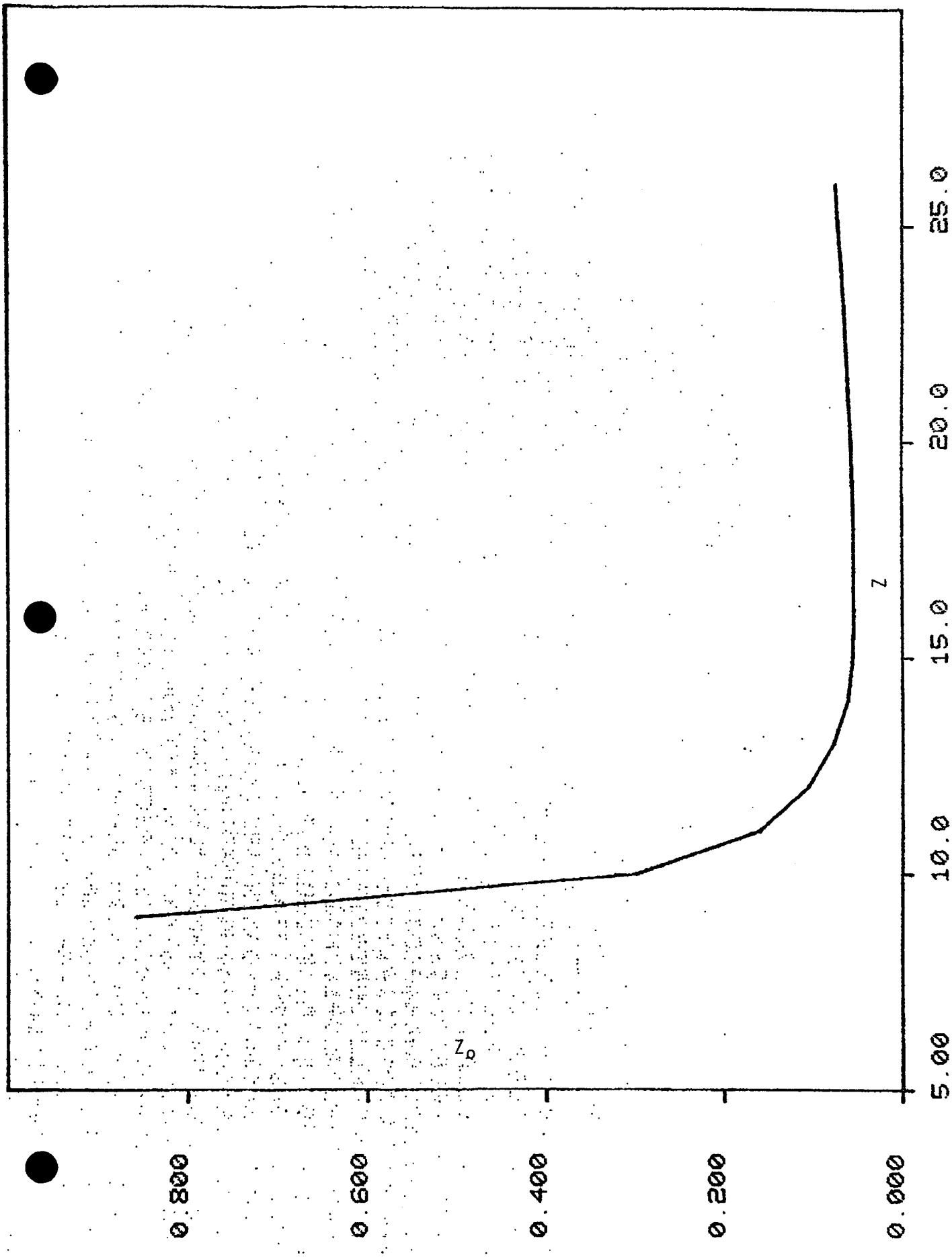


Figure 7.3a

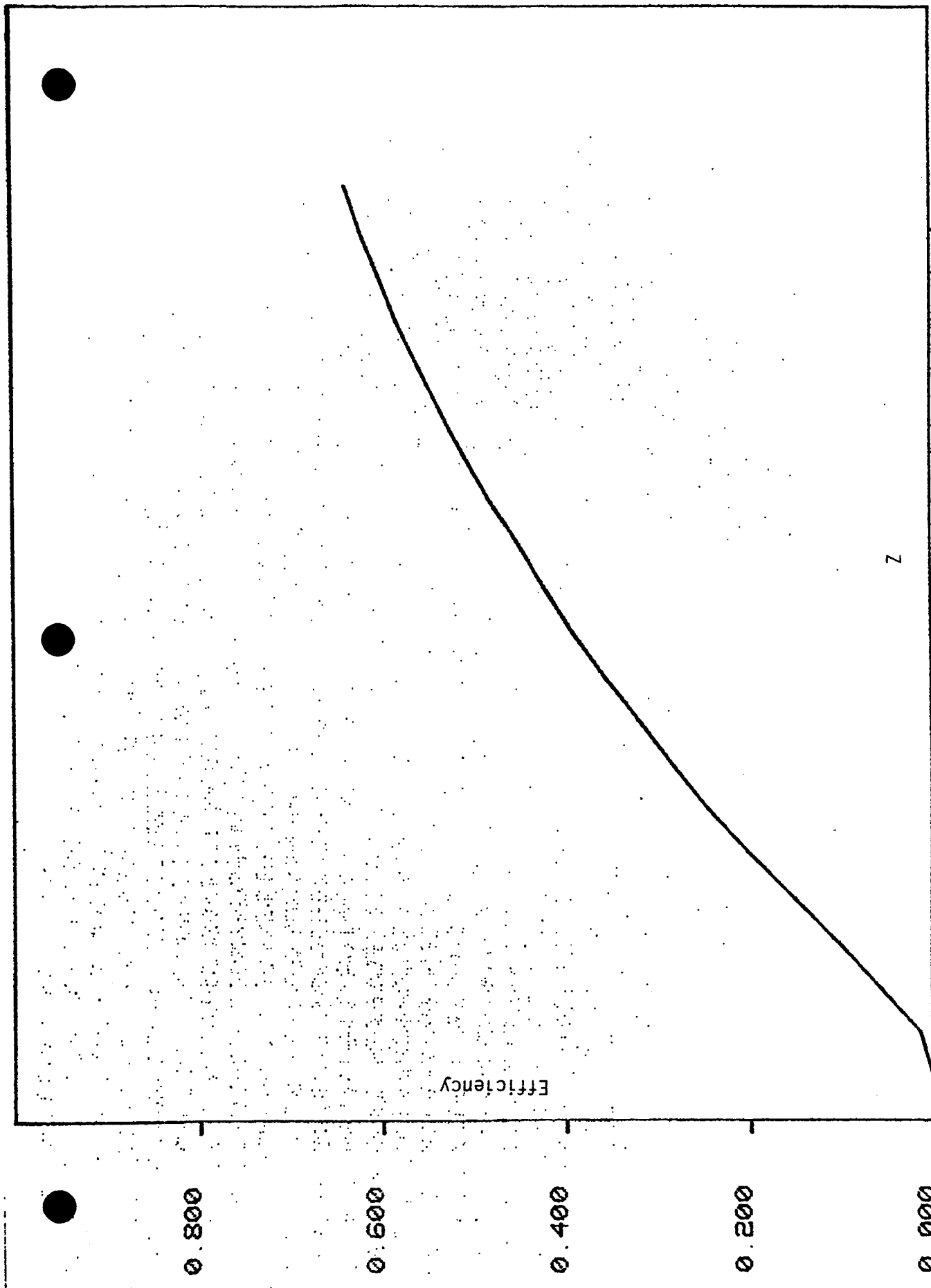


Figure 7.3b

Figure 7.4. a) The expected error in Z and b) the efficiency of a stack of intermixed Lexan and cellulose nitrate layers. The conditions are the same as in Figs. 7.1-7.3.

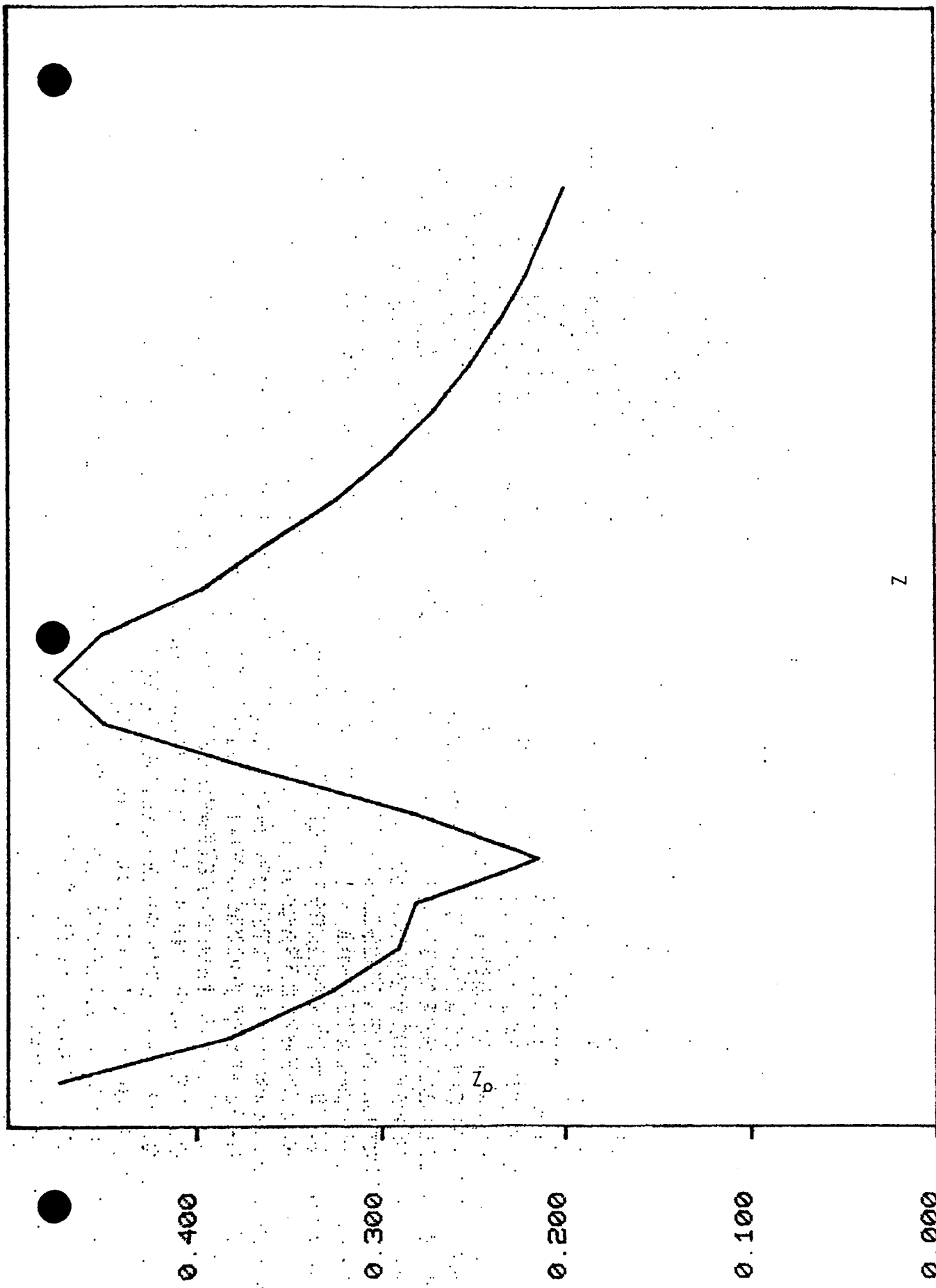


Figure 7.4a

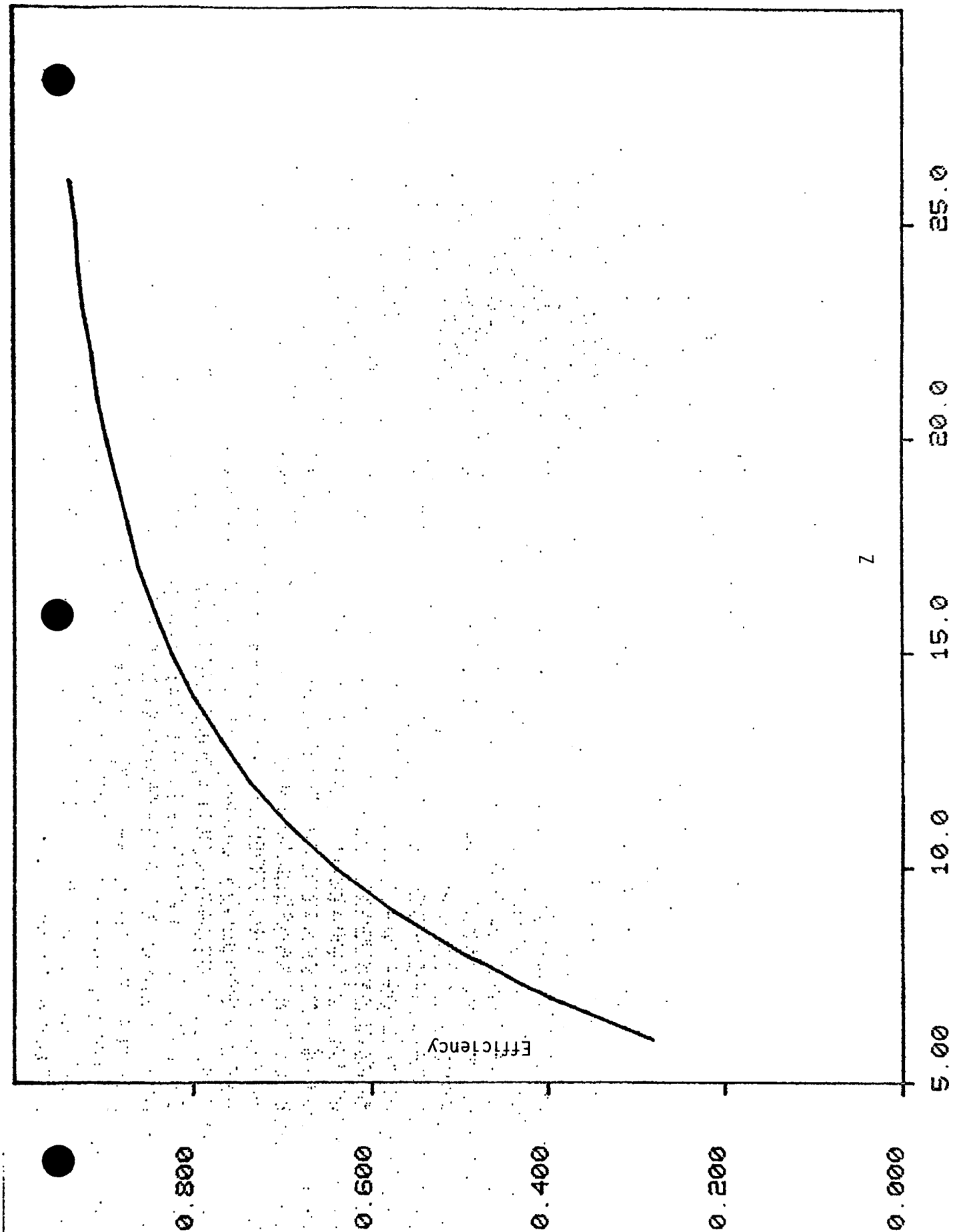


Figure 7.4b

Figure 7.5. a) The expected error in Z and b) the efficiency of a stack of intermixed Lexan, cellulose nitrate, and CR-39 layers. Each layer sequence contains two Lexan layers, one cellulose nitrate layer, and one CR-39 layer. The other conditions are the same as for Figs. 7.1-7.3.

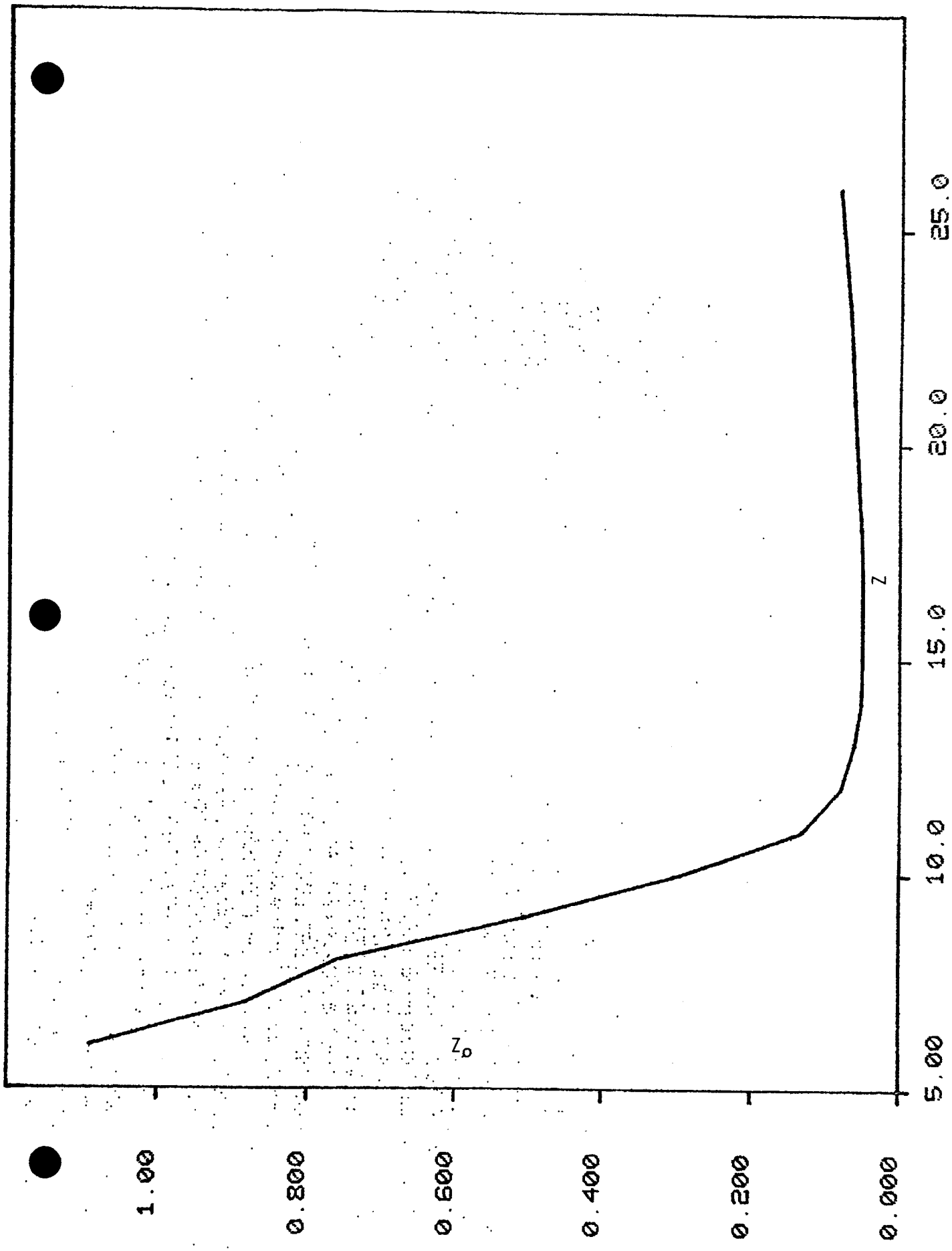


Figure 7.5a

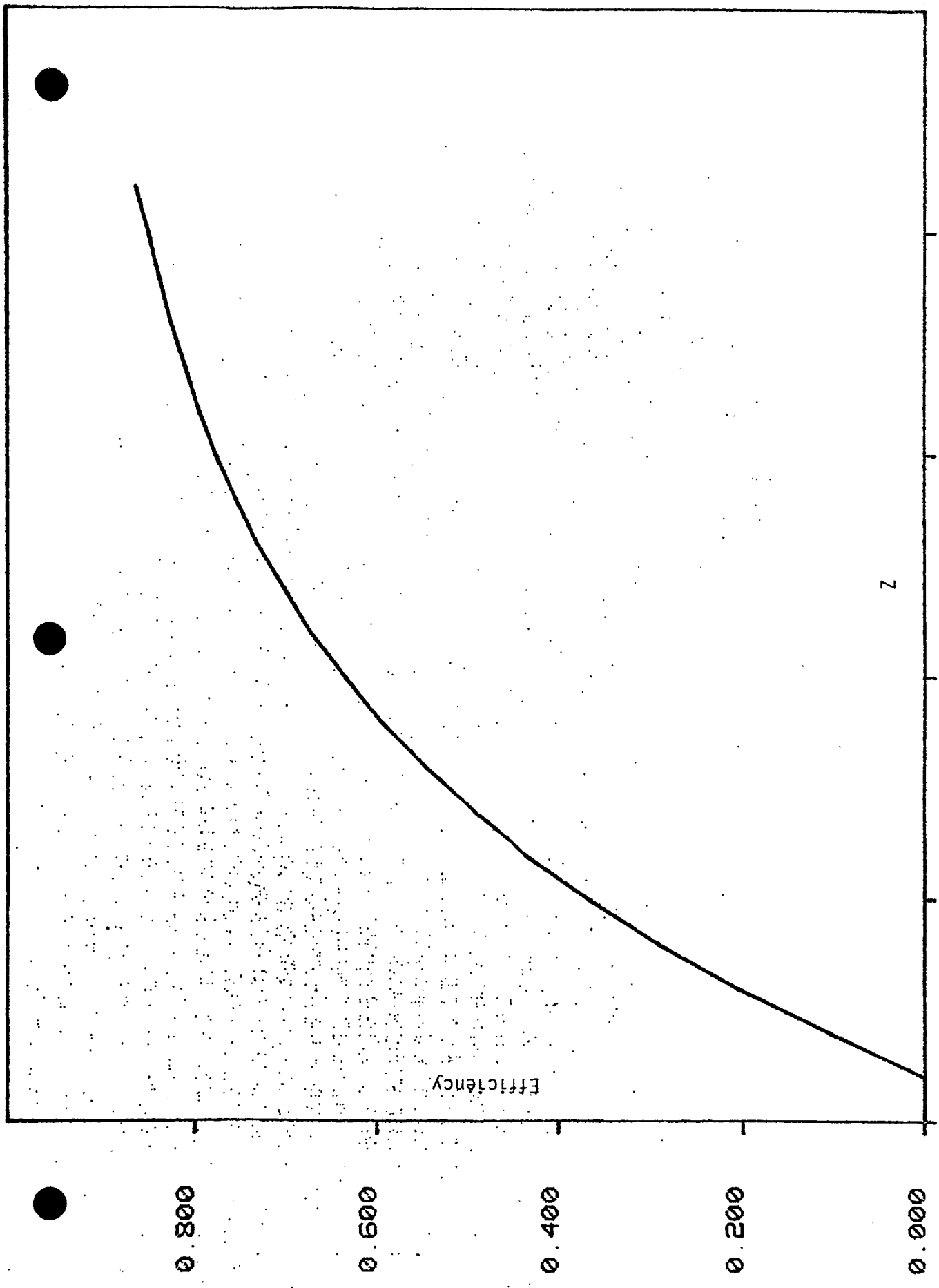


Figure 7.5b

This stack appears to perform quite well. It has, however, two drawbacks. One of these is that for intermediate Zs the Z resolution is not as good as one might hope for. The other drawback is that, since the layers are relatively thin, a large stack contains many layers. The second combination is the combination of two Lexans, one CN and one CR-39 layer. This layer sequence is repeated many times to produce the detector stack. This is shown in Fig. 7.5a,b. In the actual stack, the layer sequence would be Lexan, CN, CR-39, Lexan, CN, etc. It can be seen that this configuration corrects for the two limitations of the previous configuration in that the intermediate Zs are more accurately determined and that, since the CR-39 has a 1 mm thickness, many fewer layers will be involved in the total stack. It can be said in conclusion that the combination of two Lexan layers, one CN layer and one CR-39 layer per layer sequence provides the best choice in both accuracy and speed. It should also be mentioned that another objective was economy. This is the primary reason for having two Lexan layers. Lexan is by far the least expensive of the three materials. Also, since the CR-39 layers are thick, the total detector area is reduced.

7.2 Overview of Analysis Procedure

Before discussing the performance of the various analysis steps in detail, it is appropriate to consider the sequence of the various steps used in reducing the measurements of a stack. In the first analysis step, the data is taken as was previously outlined. An example of the raw data produced by the data-taking program is shown in Fig. 7.6 which shows the array of measured positions for a stack of CN plastic exposed to cosmic rays. It can be seen that there are two types of dot configurations. One is a set of linear patterns of dots representing high-Z particles which register in many sequential layers. These lines of dots are overlaid upon a relatively random pattern of what appears to be background dots produced by light particles which register in only

Figure 7.6. Array of measured track positions in a stack containing 85 layers of cellulose nitrate plastic track detector. The 41 mm by 24 mm area contains 45,072 measured positions. Many of these, corresponding to heavy particles which register in many layers, can be seen to be aligned in long rows.

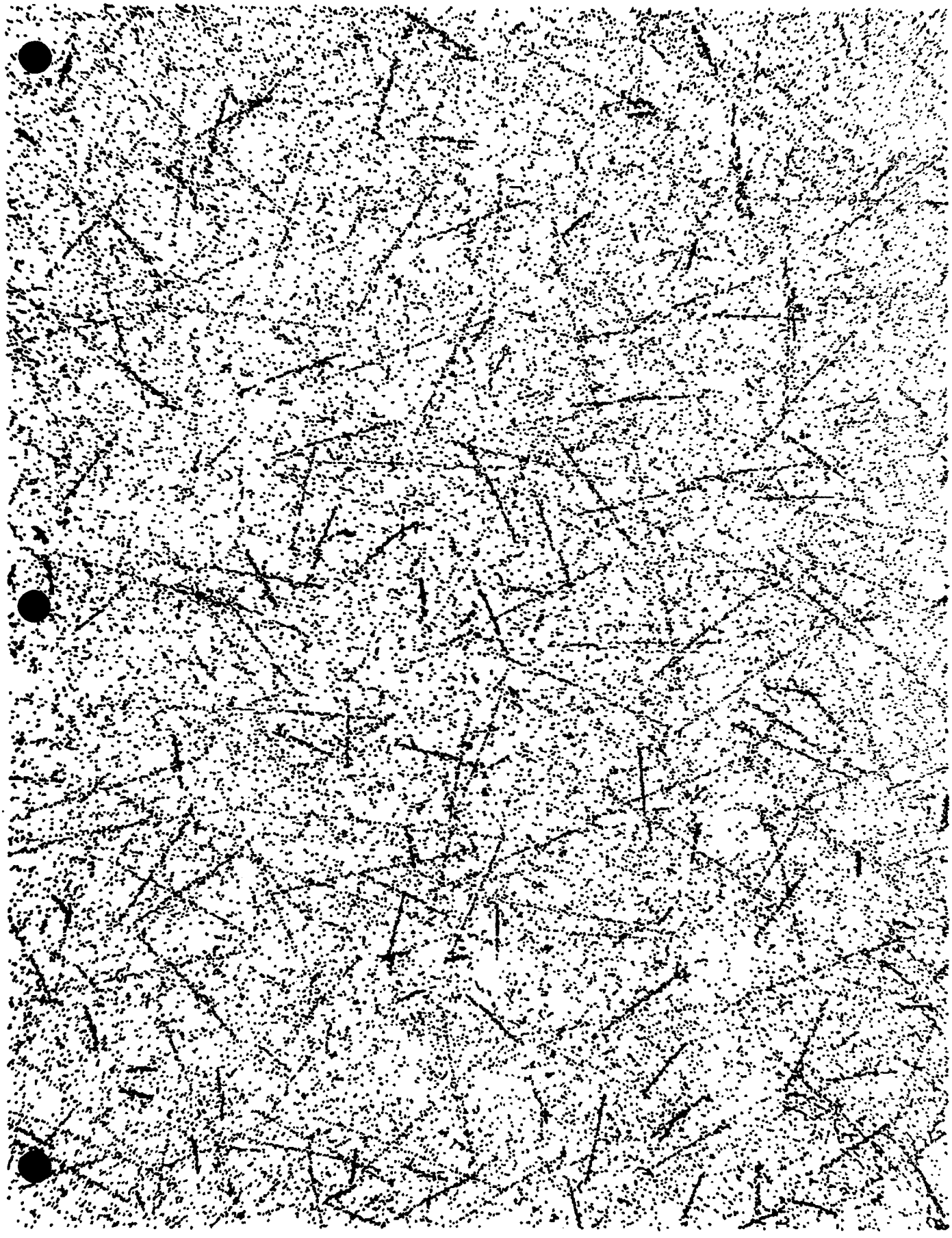


Figure 7.6

one or possibly two layers. It is the lines of dots that can be reduced to produce Z analysis data.

Figure 7.7 shows the results of the next step in the Z-analysis procedure, the tracing. In the figure, the raw data is shown with superimposed lines representing the fit trajectories as seen projected onto the x,y plane. The figure illustrates that the tracing program does, indeed, locate lines of tracks in three dimensions and fit trajectories to them.

Not only does the tracing program fit the correlated events for a given particle but it also assembles together in one contiguous section of the output file the area measurements for individual particles. An example of this data is shown in Fig. 7.8 which shows the track area versus layer number for the correlated particle event. The next step in the analysis procedure is to Z-analyze this data. Figure 7.9 shows the Z analysis of an individual particle. The fit line is shown through the data points and, at the top of the figure, the parameters as determined by the Z-analysis program are given.

The final step is to assemble Z values determined for individual particles into a Z spectrum. Such a Z spectrum is shown in Fig. 7.10.

7.3 Speed of Various Steps

The speed of the HZE analysis procedure is dependent on many parameters. The primary parameter is the number of events that must be analyzed. If we are dealing with a large stack of sensitive material and a relatively high fluence of particles such as would be experienced on a long space exposure, the time involved is much greater than would be involved for a stack with a low fluence of particles. An additional constraint on the speed is the relative size and spacing of the tracks on the layers. For example, if the tracks are very small and widely separated, a large magnification would have to be used to obtain the needed accuracy on each of the events, even though there aren't many per layer. In contrast, if we are using relatively thick layers

Figure 7.7. Tracing program performance test plot. Lines representing the fit trajectories are superimposed upon the measured track positions as given by dots. This data is from a stack of 105 CR-39 layers exposed to Fe^{56} particles at the LBL Bevalac.

Figure 7.8. Measured track opening areas as a function of the layer number for one traced particle in a stack of Lexan polycarbonate detectors exposed to cosmic rays. Note the somewhat depressed area measurement at the stopping point end of the trajectory indicating that the particle stopped within the layer.

Figure 7.9 Plot demonstrating the performance of the Z computation program, AUZC. The measured track areas, normalized to ($\frac{1}{2}$ the original layer thickness)^2, are given as Xs, and the computed values of the areas are given by the line through them. The numerical values of the track parameters are shown at the top of the plot.

Figure 7.10. Distribution of Z values measured in a cellulose nitrate stack exposed to cosmic rays.

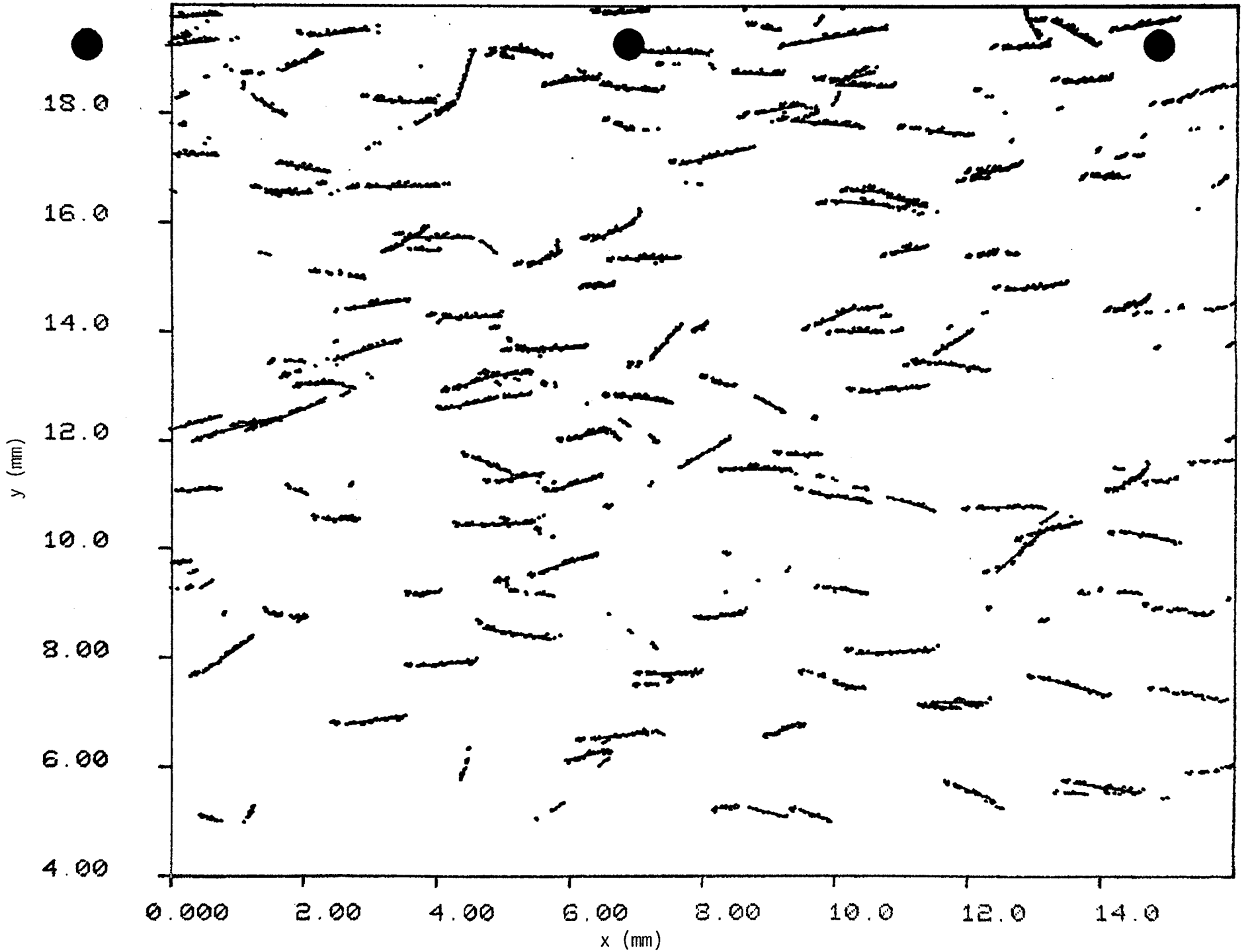

18.0
16.0
14.0
12.0
10.0
8.00
6.00
4.00
y (mm)
0.000
2.00
4.00
6.00
8.00
10.0
12.0
14.0
x (mm)

Figure 7.7

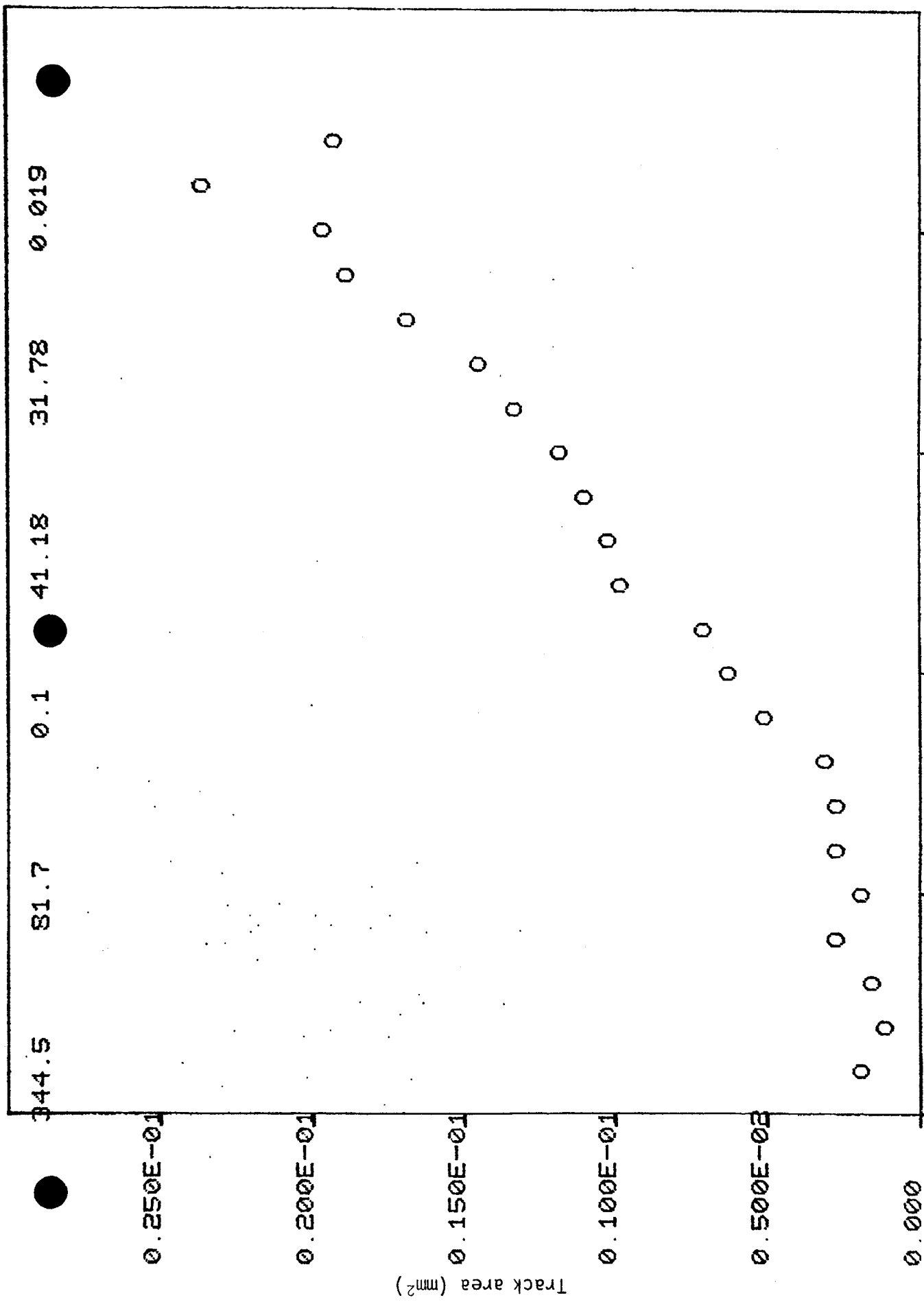

Track area (mm²)
0.250E-01
0.200E-01
0.150E-01
0.100E-01
0.500E-02
0.000
295.
300.
305.
310.
315.
Layer number
44.5
81.7
0.1
41.18
31.78
0.019

Figure 7.8

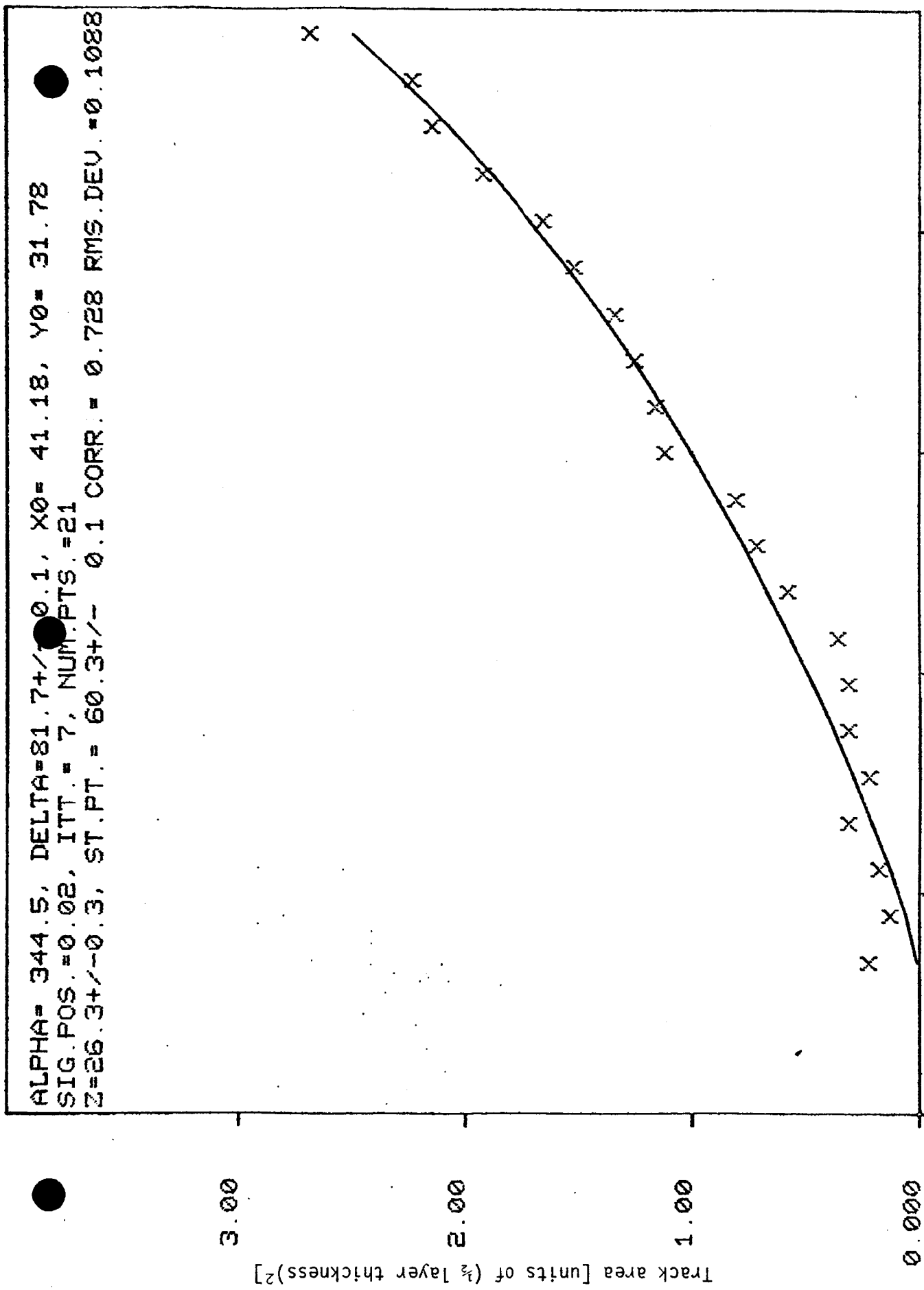

ALPHA= 344.5, DELTA=81.7+/- 0.1, X0= 41.18, Y0= 31.78
SIG.POS.=0.02, ITT.= 7, NUM.PTS.=21
Z=26.3+/-0.3, ST.PT.= 60.3+/- 0.1 CORR.= 0.728 RMS.DEV.=0.1088
Track area [units of (⅛ layer thickness)²]
3.00
2.00
1.00
0.000
55.0
56.0
57.0
58.0
59.0
Depth in stack (mm of H2O)

Figure 7.9

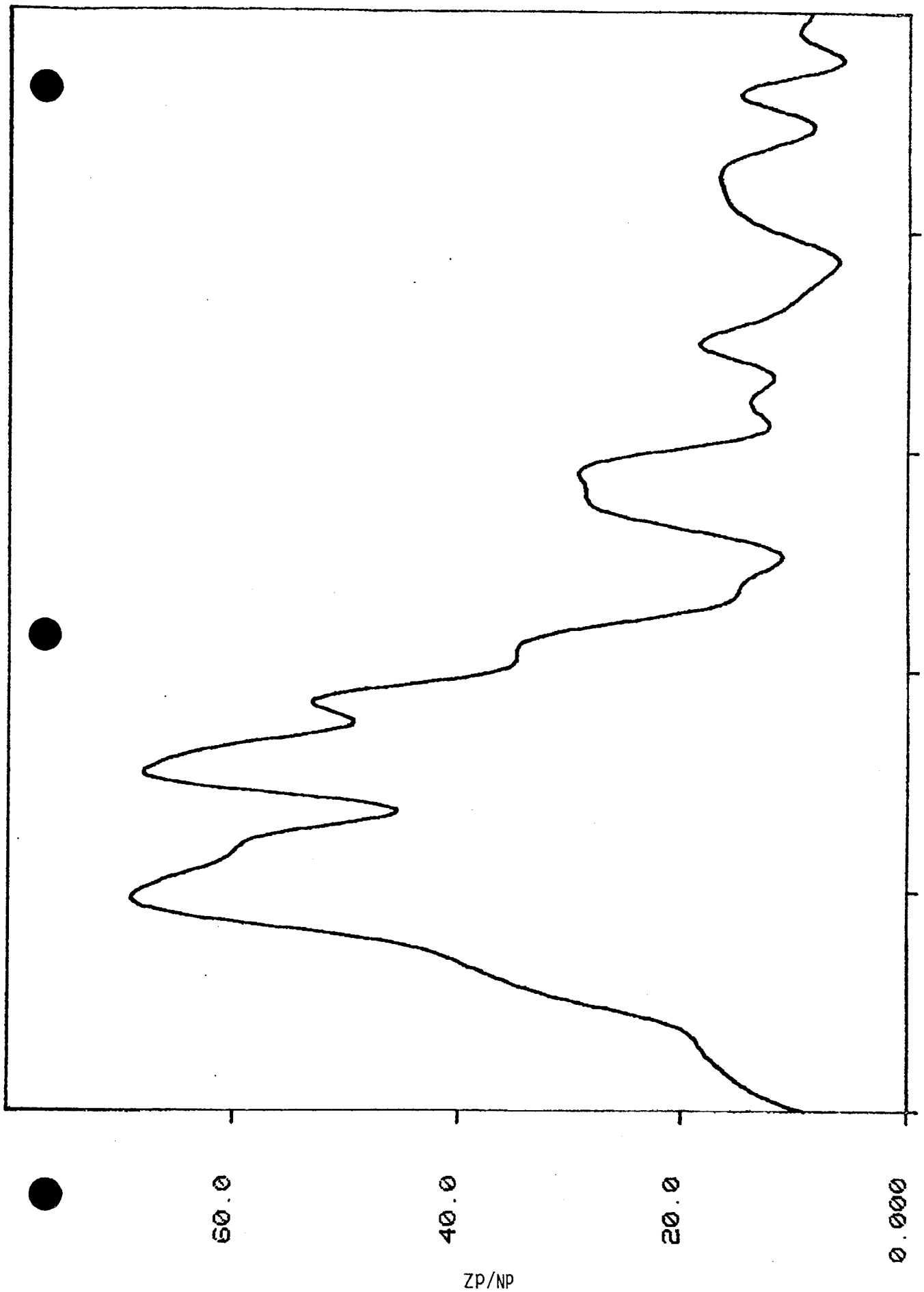


Figure 7.10

which, when etched for a ling time, produce large area tracks, low magnifications can be used and still obtain the required accuracy in area measurement. Therefore, large detector areas can be covered quickly in the analysis. In assessing the speed of data taking, the one parameter which is most significant is the fact that it takes approximately 2.5 seconds per field to measure the data. One-half second of this is the time necessary to move the stage from one field to the next; the other two seconds are consumed in processing the image for the field. Therefore, if 100 fields are required to read out one layer, 250 seconds will be required. At present, with the layer types and the fraction of the layer removed as previously indicated, scale factors of from 50-150 picture points per mm are required to obtain the necessary accuracy in area measurement. With the full field size of 3×10^5 picture points, this then translates into areas of from 14 mm^2 to 120 mm^2 or 1.2 cm^2. Therefore, it can be said that the typical speed of the data taking is approximately from 3 to 30 cm^2 per minute. This means that a typical layer of approximately 25 cm^2 can take from one to ten minutes. Another way of stating the measurement speed is that for the standard layer configuration, the rate of data taking is approximately 1 cm^2 of stack per minute. For a relatively large stack of, say, the dimensions of 10 cm by 10 cm by 10 cm, this would then require 1000 minutes or approximately two working days of time for readout. It should be stated, however, that the numbers just given are quite approximate and there is a great dependence on the precise conditions of the etch and the exposure.

It is also possible to use the automated HZE procedure for radically different types of exposures. For example, a very high exposure of particles could be handled if very thin detector layers were used with relatively small stacks. These detectors, in this case, would still be etched to remove a fairly large fraction of the layer but it would be a short time producing small area tracks.

The data would then be read out under the microscope rather than with the macro viewing device. Very small step sizes would be used between adjacent fields and relatively small area would be detected. In the opposite extreme, very low fluences of tracks on very large detector areas could be detected by using thick layers which could be viewed at relatively low magnification. In fact, with sufficient thickness, a whole detector layer could be digitized in one field. The main objective in assessing the types of layers used, that is, the thickness and type of material, is to match the Z of the particle to the type of layer used. For instance, if a low-fluence situation is involved with relatively low-Z particles, quite sensitive detector layers would be required, such as CR-39, so that the sufficient number of measurements would be available along each particle track. If we were detecting the same Z particle in a high-fluence environment using thin layers, the detector would need to be somewhat less sensitive in that it would register in several layers and the areas would decrease along the trajectory with a much less sensitive material where the layers are thin.

7.4 Tracing Accuracy

The primary indication of the accuracy of the trajectories, which directly manifests the accuracy in the measurement, is the RMS deviation of the measured points about the fit trajectories obtained by the tracing program. In Fig. 7.11 the distribution of the rms values of the deviations about a typical trajectory are shown. It can be seen that the typical accuracy in our standard reduction set-up in which the scale factors for the measurement process are from 50 to 150 picture points per mm, are of the order of 10-30 μm. The errors about the fit trajectory just mentioned lead directly through the least-squares trajectory analysis into an error in the dip angle. The distribution of such errors for typical particles is shown in Fig. 7.12. Of course, there are also

Figure 7.11. Distribution of the RMS deviation of the measured track positions about the trajectories of 125 particles in a Lexan cosmic-ray stack. It should be noted that the peak in the distribution occurs at approximately 20 μm, which is only 11% of the detector layer thickness.

Figure 7.12. Distribution of the error in the measured particle dip angles in a Lexan cosmic-ray stack. This is a worst case situation in that Lexan is the least sensitive of the detector materials used and therefore yields the fewest measurements per particle.

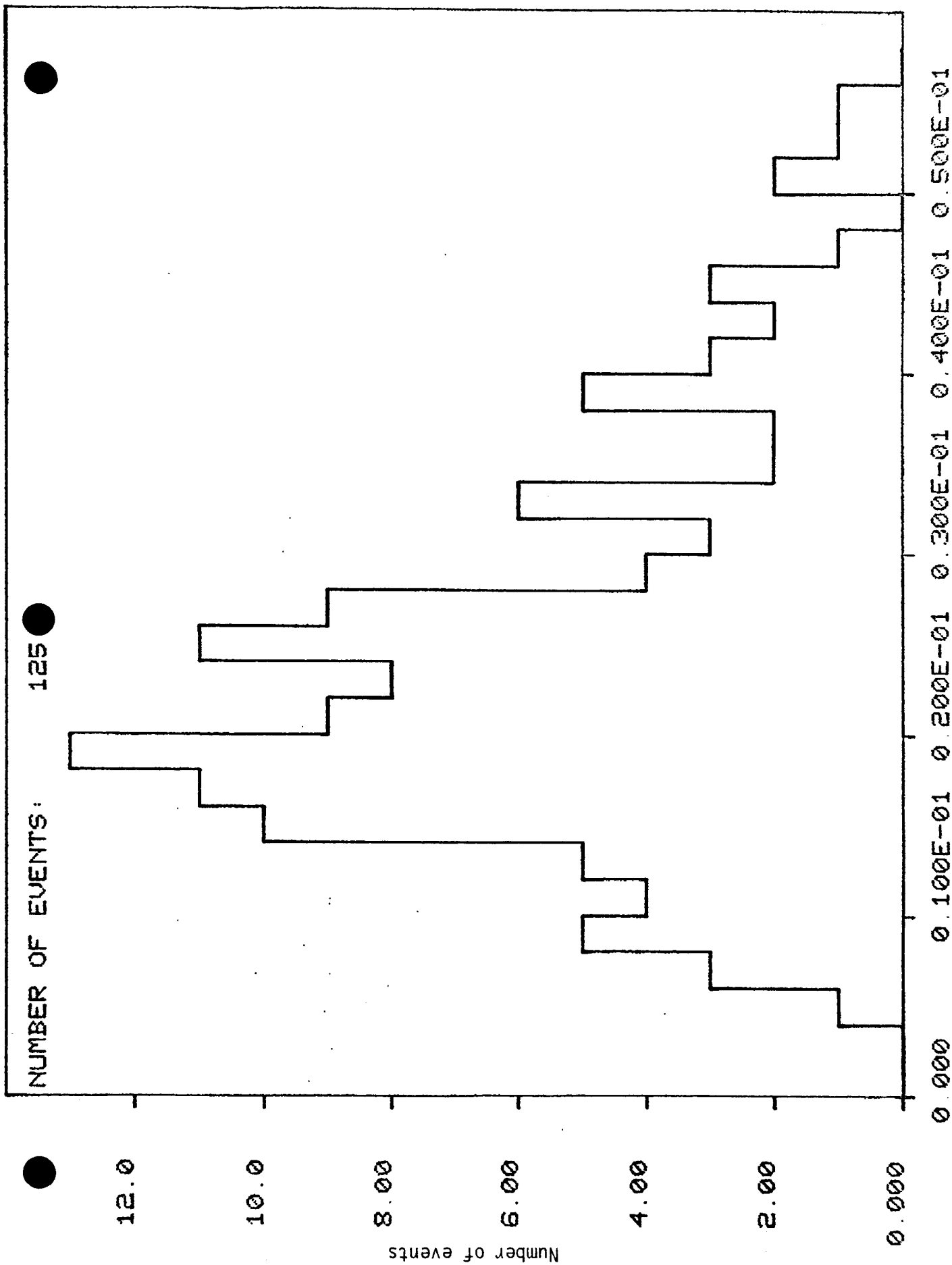

NUMBER OF EVENTS: 125
Number of events
12.0
10.0
8.00
6.00
4.00
2.00
0.000
0.000
0.100E-01
0.200E-01
0.300E-01
0.400E-01
0.500E-01
RMS error of track position (mm)

Figure 7.11

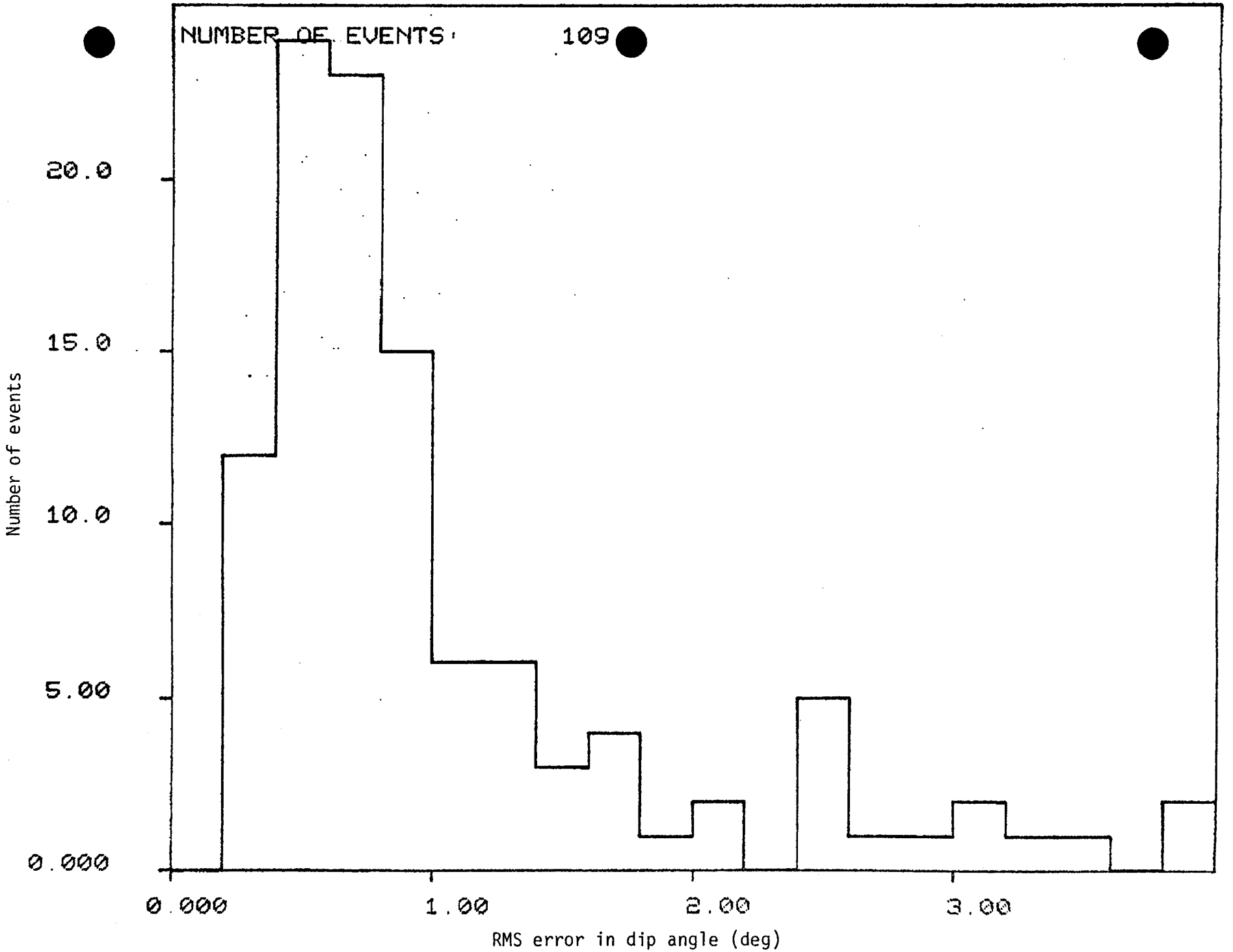

NUMBER OF EVENTS: 109
20.0
15.0
10.0
5.00
0.000
Number of events
0.000
1.00
2.00
3.00
RMS error in dip angle (deg)

Figure 7.12

errors introduced in the other parameters of the trajectory but they have no impact on the Z analysis. Therefore, the error in the dip angle is the primary one of interest. It can be seen that the dip-angle errors for particles which are sufficiently heavy to produce several measurements along the trajectory are quite small. This means that the uncertainty in the dip angle does not contribute appreciably to the error in the Z as determined in the Z analysis.

7.5 Z Analysis

The primary indication of the performance of the Z analysis is the accuracy of the Z and the ability of the program to represent the data. As was already seen in the preliminary description of the analysis procedure, the program does accurately represent the data. Since the Z-analysis program is an iterative least-squares process, the only limitation to the representation faithfulness is the correctness of the model. In a large number of tests, the model has proven to be quite adequate in representing the track areas. The accuracy in Z can be assessed by examining the histogram of the distribution of the errors in Z for the composite Z spectrum from a stack. Such a distribution is shown in Fig. 7.13, which shows the errors in Z as determined by the rms deviation of the points about the fit. Another method for assessing the Z accuracy is to examine the Z spectrum from an exposure to a known spectrum of particles. An example of such a spectrum is shown in Fig. 7.14, which shows the Z spectrum obtained from an iron exposure in a CR-39 stack. From both Fig. 7.13 and Fig. 7.14 it can be seen that typical Z accuracies of better than ± 1 in Z are obtainable.

Another criterion for assessing the Z-analysis process is to state the range of possible Z values detectable. Both the parametric analysis program and experience show that the possible range of Zs is from carbon particles, that is $Z = 6$, to iron particles, $Z = 26$. It is quite possible to detect higher

Figure 7.13. Distribution of the error in Z for particles from a Lexan cosmic-ray stack. These errors are inferred from the RMS deviation of the measured track areas about the fit curve (as shown in Fig. 7.9).

Figure 7.14. Z spectrum measured in a CR-39 stack exposed to Fe^{56} particles at the LBL Bevalac. The narrowness of the peak produced by the primary particles indicates the quality of the Z resolution. A number of secondary particles, produced through fragmentation, are seen as well.

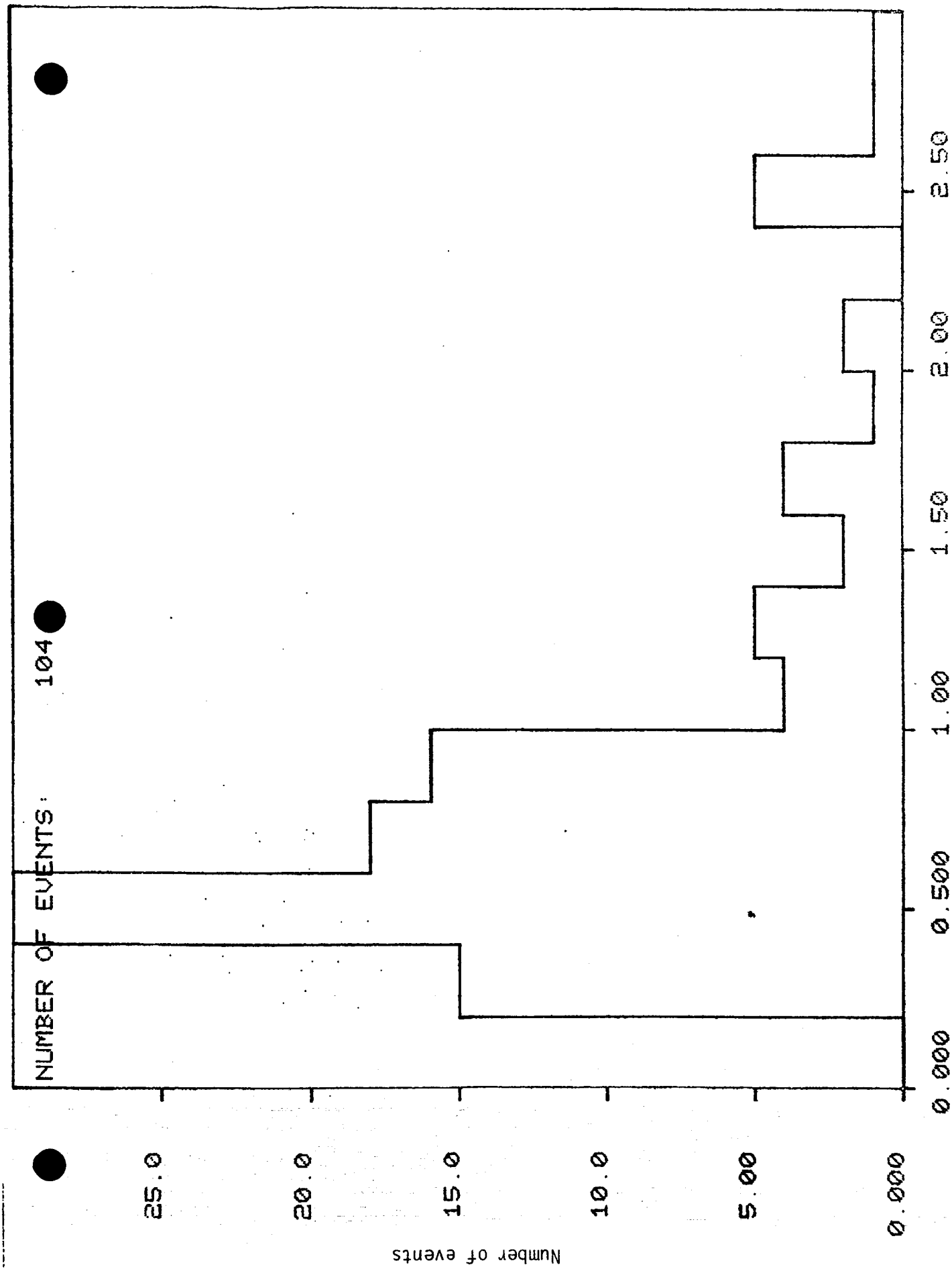

NUMBER OF EVENTS: 104
Number of events
25.0
20.0
15.0
10.0
5.00
0.000
0.000
0.500
1.00
1.50
2.00
2.50
RMS error in Z

Figure 7.13

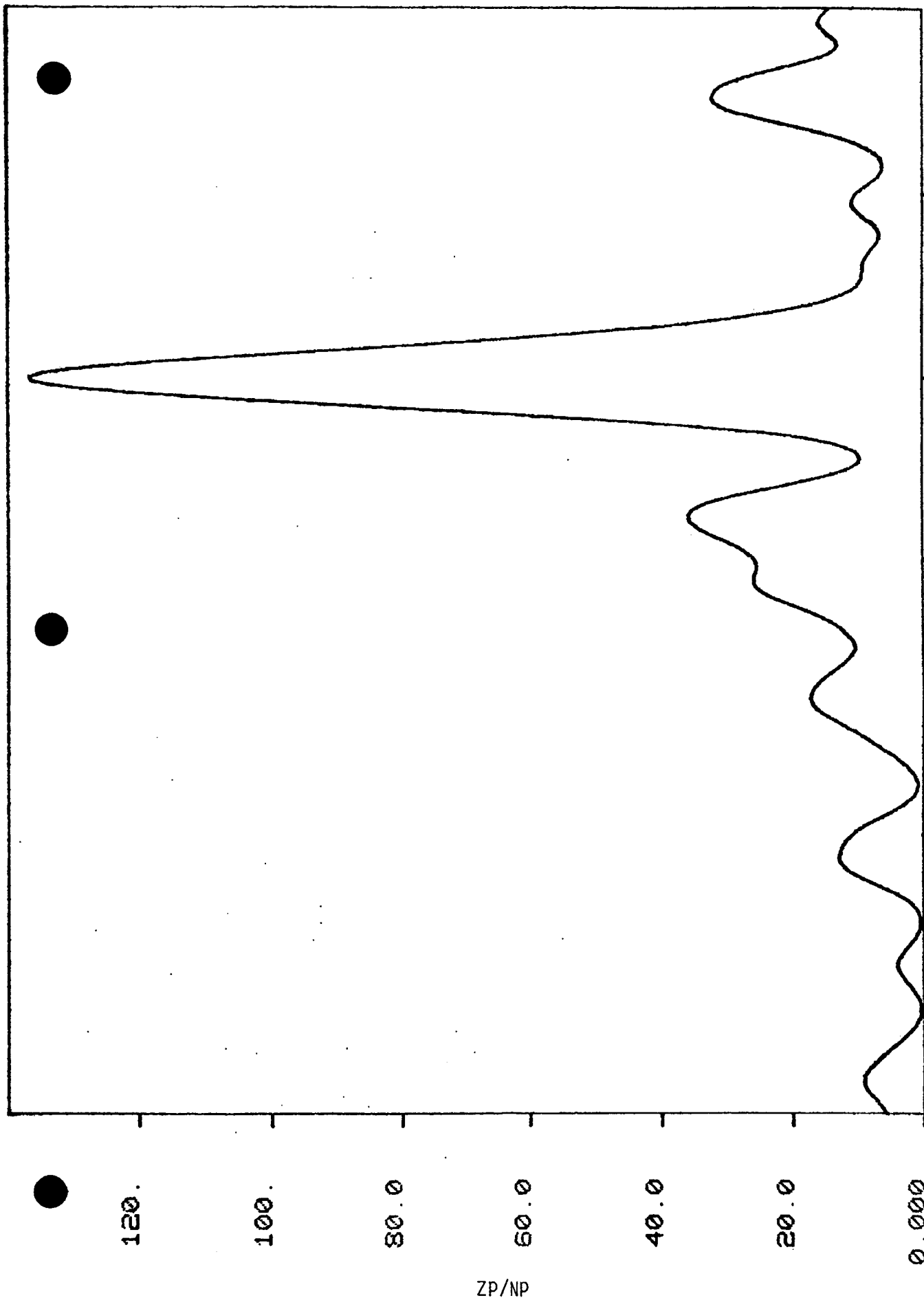


Figure 7.14

Z values using relatively insensitive detector layers, however, in practice, this is academic because the normal cosmic-ray spectrum contains such a small fluence of these particles that very few if any of such particles intersect a reasonably sized stack.

7.6 Statistical Output

An analysis of the performance for the statistical output consists of stating the limits of the particle fluences and the energies that can be detected. With the standard measurement configuration using the macrostage, the upper fluence limit is approximately 100 particles/cm^2. For fluences greater than this, the position measurements are not sufficiently accurate to allow successful tracing of the particles. The minimum fluence is established by the accuracy requirements and the detector area. For instance, if 100 cm^2 of detector area are available and 10% counting statistics are required, the lower fluence limit would be 1 particle/cm^2. This then places limits on the time aloft. In practice, the upper limit is not a limit if different readout techniques are used as was previously described.

The limits in particle energy are determined, in part, by the size of the detector stack and, in part, by the Z of the particle. For very heavy particles such as iron, even relativistic particles are detectable. Therefore, if one relaxes the criterion that the particle must stop in the stack, which is possible if reduced accuracy in the Z determination is acceptable, then it is possible to go to almost relativistic particles in Z. However, for lighter particles, it is necessary for the particle to stop in the stack. Therefore, in practice, the real constraint on the energy range of the particle which can be analyzed in a stack is the size of the stack. The minimum energy is established by requiring several layers on the surface to detect the particle. This implies minimum particle energies of several tens of MeV/nucleon. The upper

energy detectable is the energy of a particle which traverses the entire stack and then stops near the exit side of the stack. Particles of this character typically have energies of several hundred MeV/nucleon. Several variations of these principles are possible however. If very large stacks are involved, higher energies can be detected. In addition, if external shielding is incorporated in the analysis, the spectrum of particles exterior to this external shielding can be analyzed when the energy of the particle is much greater.

7.7 Tests of System

Two different types of system tests have been performed to date. These are exposures made in space and accelerator exposures. The accelerator exposures are the most informative in that the Z spectrum is known. An example of such a Z spectrum is shown in Fig. 7.14 which gives the distribution of Z values for an iron exposure. Space exposures show the resolution of particles in adjacent Z values. An example of such a spectrum is shown in Fig. 7.15. In such an exposure typically the alternate even Zs produce peaks with the odd Zs not showing isolated peaks. This is consistent with the accuracy of approximately ±1 in Z which is not sufficient to analyze adjacent particle peaks and the fact that the even Zs are much more abundant in the cosmic-ray spectrum than the odd Z values.

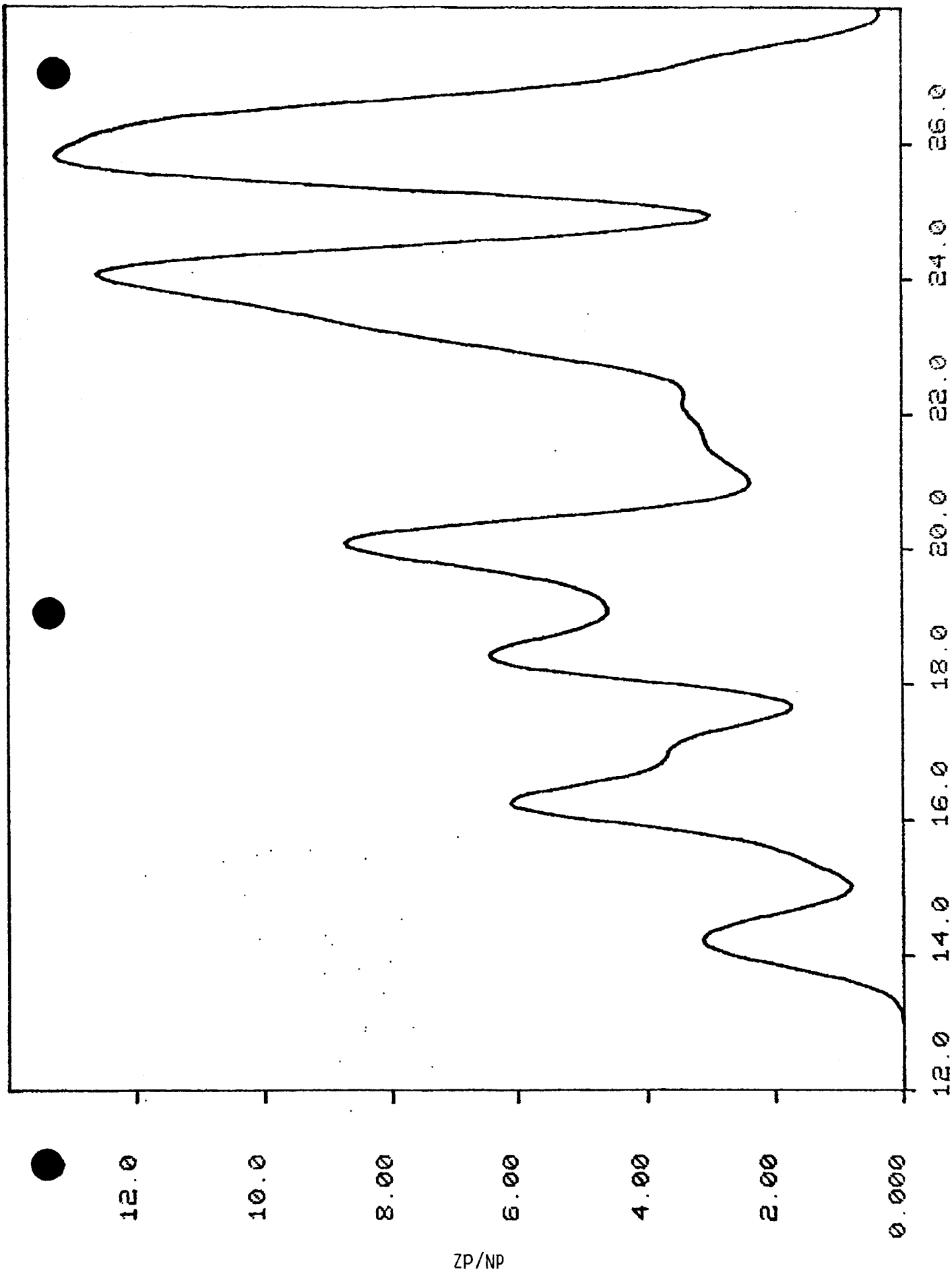


Figure 7.15. Z spectrum measured in a Lexan stack exposed to cosmic rays.

8. SUMMARY

The automated HZE system is a flexible system capable of producing automated Z and LET spectra from either space or accelerator exposures to particles. It produces results quickly and reasonably accurately in a time which is orders of magnitude shorter than the time required to produce manual measurements of the same quantities. The accuracy of the system is quite adequate for dosimetric purposes.

There are some limitations to the automated HZE process. None of these appears to be inherent in the nature of the automated HZE process itself, so it is expected that further development can improve various aspects of the procedure such as the accuracy, speed, and fluence limitations.

AUTOMATED ANALYSIS SYSTEM FOR HZE PARTICLE TRACK DOSIMETERS

APPENDIX

```
        .TITLE ASUM
;FORTRAN CALLABLE SUBROUTINE TO FORM THE SUM OF GRAY VALUES FROM
;A 16 BY 16 ARRAY OF POINTS IN REFRESH MEMORY STARTING AT THE
;ADDRESS PAIR IX,IY.  THE SUM, DIVIDED BY 8 IS OUTPUT THROUGH THE
;ARGUMENT ISUM.  THE TYPICAL CALL IS
;       CALL ASUM(IX,IY,ISUM)
        .MCALL .REGDEF
        .REGDEF
        .GLOBL ASUM
ASUM:   TST (R5)+               ;INCREMENT ARGUMENT POINTER
        MOV @(R5)+,R0           ;X ADDRESS
        MOV #16.,R2             ;X COUNTER
        CLR @2(R5)              ;CLEAR SUM
LOOP1:  MOV #16.,R3             ;Y COUNTER
        MOV @0(R5),R1           ;Y ADDRESS
        CLR R4                  ;CLEAR INTERMEDIATE SUM
        MOV R0,@#164204         ;LOAD X ADDRESS
        TST (R0)+               ;INCREMENT X ADDRESS
LOOP2:  MOV R1,@#164206         ;LOAD Y ADDRESS
        TST (R1)+               ;INCREMENT Y ADDRESS
        ADD @#164236,R4         ;CREATE INTERMEDIATE SUM
        DEC R3                  ;DECREMENT COUNTER
        BNE LOOP2               ;CONTINUE
        ASR R4                  ;DIVIDE BY 2 THREE TIMES
        ASR R4
        ASR R4
        ADD R4,@2(R5)           ;ADD TO SUM
        DEC R2                  ;DECREMENT X COUNTER
        BNE LOOP1               ;CONTINUE
        RTS PC                  ;RETURN
        .END ASUM
```

```
C       AUALS.FOR
C       SELECTION OF AUZC OUTPUT BY ALPHA.
        DIMENSION P1(6),P2(6),Z(100),A(100),AC(100)
        DPR=45./ATAN(1.)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 3
3       FORMAT(' ALPHA RANGE')
        ACCEPT 4,D1,D2
4       FORMAT(7F10.0)
        D1=D1/DPR
        D2=D2/DPR
5       READ(1,END=6)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        IF(P1(1).LE.D1.OR.P1(1).GT.D2)GO TO 5
        WRITE(2)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        GO TO 5
6       END
```

```
C       AUCALP.FOR
C       PLOT OF CALIBRATION LAYER DATA
        DIMENSION S(4)
        DATA S/1.E37,-1.E37,1.E37,-1.E37/
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 5
5       FORMAT(' MAXIMUM AREA')
        ACCEPT 6,AM
6       FORMAT(7F10.0)
2       READ(1,END=3)X,Y,J,A
        IF(A.GT.AM)GO TO 2
        S(1)=AMIN1(S(1),X)
        S(2)=AMAX1(S(2),X)
        S(3)=AMIN1(S(3),A)
        S(4)=AMAX1(S(4),A)
        GO TO 2
3       CALL SCALE(S)
        REWIND 1
4       READ(1,END=7)X,Y,J,A
        CALL BPOINT(S,X,A)
        GO TO 4
7       END
```

```
C       AUCL.FOR
C       CLEANING UP AUHZE DATA BY NEGATING AREA MEASUREMENTS OF
C       UNDESIREABLE POINTS.
        DIMENSION L(50),CH(4),Z(50),A(50),S(4)
        DATA CH/-.5,72.5,-.5,35.5/
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'B',-1,'NEW')
4       READ(1,END=3)AL,DE,SD,X0,Y0,SIG,N,(L(I),A(I),I=1,N)
        S(1)=1.E10
        S(2)=-S(1)
        S(3)=0.
        S(4)=0.
        DO 5 I=1,N
        Z(I)=L(I)
        S(1)=AMIN1(Z(I),S(1))
        S(2)=AMAX1(Z(I),S(2))
5       S(4)=AMAX1(S(4),ABS(A(I)))
        IF(N.LT.3)GO TO 4
        S(4)=1.1*S(4)
        CALL SCALE(S)
        DO 9 I=1,N
        CALL LET(S,Z(I),ABS(A(I)))
        IF(A(I).GT.0.)TYPE 10,79
9       IF(A(I).LT.0.)TYPE 10,88
10      FORMAT(1X,72A1)
        CALL LET(CH,4.,35.)
        TYPE 11,57.3*AL,57.3*DE,57.3*SD,X0,Y0,SIG
D       CALL VECTOR(CH,8.,34.,72.,34.)
11      FORMAT(1X,3F10.1,2F10.2,F10.3)
6       CALL HAIRS(S,X,Y,ICHAR)
        IF(ICHAR.NE.78)GO TO 7
        DM=1.E10
        DO 8 I=1,N
        D=(ABS(A(I))-Y)/(S(4)-S(3))
        DP=(Z(I)-X)/(S(2)-S(1))
        D=D*D+DP*DP
        IF(D.GT.DM)GO TO 8
        DM=D
        J=I
8       CONTINUE
        CALL LET(S,Z(J),A(J))
        TYPE 10,88
        A(J)=-A(J)
        GO TO 6
7       WRITE(2)AL,DE,SD,X0,Y0,SIG,N,(L(I),A(I),I=1,N)
        GO TO 4
3       END
```

```
C       AUCNT.FOR
C       COUNTING NUMBER OF PARTICLES IN AUZC OUTPUT FILE.
        TYPE 1
1       FORMAT(' FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        N=0
2       READ(1,END=3)X
        N=N+1
        GO TO 2
3       TYPE 4,N
4       FORMAT(I20)
        END
```

```
C       AUCOMB.FOR
C       COMBINING AUZC OUTPUT FILES.
        DIMENSION P1(6),P2(6),Z(100),A(100),AC(100)
        TYPE 5
5       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 6
6       FORMAT(' NUMBER OF INPUT FILES')
        ACCEPT 1,N
1       FORMAT(I10)
        DO 2 I=1,N
        TYPE 7
7       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 8
8       FORMAT(' MINIMUM NUMBER OF TRACKS/PARTICLE')
        ACCEPT 1,MIN
4       READ(1,END=3)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        IF(M.LT.MIN.OR.Z(M)-Z(1).GT.20.)GO TO 4
        WRITE(2)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        GO TO 4
3       CALL CLOSE(1)
2       CONTINUE
        END
```

```
C       AUCTOL.FOR
C       CONVERSION OF CALIBRATION STRIP DATA TO LEAST SQUARES
C       FIT FORMAT
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
3       READ(1,END=4)X,Y,I,A
        WRITE(2)X,A
        GO TO 3
4       END
```

```
C       AUDIPS.FOR
C       SELECTION OF AUZC OUTPUT BY DIP ANGLE
        DIMENSION P1(6),P2(6),Z(100),A(100),AC(100)
        DPR=45./ATAN(1.)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 3
3       FORMAT(' DIP ANGLE RANGE')
        ACCEPT 4,D1,D2
4       FORMAT(7F10.0)
        D1=D1/DPR
        D2=D2/DPR
5       READ(1,END=6)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        IF(P1(2).LE.D1.OR.P1(2).GT.D2)GO TO 5
        WRITE(2)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        GO TO 5
6       END
```

```
C       AUDIRS.FOR
C       SELECTION OF AUZC OUTPUT BY DIRECTION OF TRAVEL
        DIMENSION P1(6),P2(6),Z(100),A(100),AC(100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 3
3       FORMAT(' DIRECTION OF TRAVEL')
        ACCEPT 4,DIR
4       FORMAT(7F10.0)
5       READ(1,END=6)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        DM=P2(3)*2.-Z(1)-Z(M)
        IF(DM.GT.0..XOR.DIR.GT.0.)GO TO 5
        WRITE(2)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        GO TO 5
6       END
```

```
C       AUDISS.FOR
C       SELECTION OF AUZC OUTPUT BY DISTANCE OF THE STOPPING POINT.
        DIMENSION P1(6),P2(6),Z(100),A(100),AC(100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 3
3       FORMAT(' DISTANCE OF STOPPING POINT RELATIVE TO THE SPAN
     * OF THE TRACKS')
        ACCEPT 4,DIS
4       FORMAT(7F10.0)
5       READ(1,END=6)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        IF(AMIN1(ABS(Z(1)-P2(3)),ABS(Z(M)-P2(3))).GT.
     *DIS*ABS(Z(M)-Z(1)))GO TO 5
        WRITE(2)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        GO TO 5
6       END
```

```
C       AUEFF3.FOR
C       EFFICIENCY CALCULATION FOR A SEMI-INFINITE STACK COMPOSED OF
C       REPEAT UNITS MADE UP OF CN, LEXAN, AND CR-39 LAYERS.
C       THE EFFICIENCY IS THE FRACTION OF THOSE PARTICLES STOPPING IN THE
C       STACK WHICH PRODUCE AT LEAST THREE MEASURABLE HOLES.
        DIMENSION S(4)
        COMMON NL(3),FR(3),H(3),F(3),RR,V0(3),CLET(3),PSI(3)
        DATA V0/0.,0.,1./,CLET/35.56,259.1,157./,PSI/2.425,2.104,1.75/,
     *F/1.302,1.117,1.27/,PI/3.1415927/
        TYPE 10
10      FORMAT(' Z INTERVAL')
        ACCEPT 2,IZI
8       TYPE 1
1       FORMAT(' IN CN: NUMBER OF LAYERS/UNIT, LAYER THICKNESS (MM),'/
     *,' FRACTION OF LAYER REMOVED DURING ETCH')
        ACCEPT 2,NL(1),H(1),FR(1)
2       FORMAT(I10,2F10.0)
        TYPE 3
3       FORMAT(' IN LEXAN: NUMBER OF LAYERS/UNIT, LAYER THICKNESS (MM),'/
     *,' FRACTION OF LAYER REMOVED DURING ETCH')
        ACCEPT 2,NL(2),H(2),FR(2)
        TYPE 18
18      FORMAT(' IN CR-39: NUMBER OF LAYERS/UNIT, LAYER THICKNESS(MM),'/
     *,' FRACTION OF LAYER REMOVED DURING ETCH')
        ACCEPT 2,NL(3),H(3),FR(3)
        RR=0.
        DO 4 I=1,3
4       RR=RR+NL(I)*H(I)*F(I)
        DO 5 I=1,3
5       H(I)=F(I)*H(I)/RR
        S(1)=6.
        S(2)=28.
        S(3)=0.
        S(4)=.99
        CALL SCALE(S)
        DO 6 IZ=6,26,IZI
        A=.25*PI
        DA=.5*A
        Z=IZ
7       CALL EFF(Z,A,E)
        IF(E.LT..5)GO TO 11
        A=A-DA
        GO TO 12
11      A=A+DA
12      DA=.5*DA
        IF(DA.GT..001)GO TO 7
        Y=1.-SIN(A)
        IF(IZ.NE.6)CALL VECTOR(S,ZL,YL,Z,Y)
        ZL=Z
6       YL=Y
        ACCEPT 2,I
        GO TO 8
        END
C
C
        SUBROUTINE EFF(Z,DELTA,E)
C
        DIMENSION R(3)
        COMMON NL(3),FR(3),H(3),F(3),RR,V0(3),CLET(3),PSI(3)
        CALL RREG(Z,DELTA,R)
        SD=SIN(DELTA)
        DO 1 I=1,3
1       R(I)=R(I)*SD
        E=0.
```

```
      EL=0.
      DO 2 I=1,3
2     EL=EL+NL(I)*R(I)
      IF(EL.GE.3.5)E=1.
      RETURN
      END
C
C
      SUBROUTINE RREG(Z,DELTA,R)
C
      DIMENSION R(3)
      COMMON NL(3),FR(3),H(3),F(3),RR,V0(3),CLET(3),PSI(3)
      PB2=1.570796
      RZ=27.39*Z**3.767/RR
      DO 1 I=1,3
      R(I)=0.
      SD=SIN(DELTA)
      IF(DELTA.GE.PB2)GO TO 2
      CD=COS(DELTA)
      TH=2.*ATAN(FR(I)*SD/CD)-DELTA
      IF(TH.LE.0.)GO TO 1
      IF(DELTA+TH-PB2)4,4,2
4     ST=SIN(TH)
      GO TO 3
2     ST=FR(I)*SD
3     ALET=CLET(I)*(1./ST-V0(I))**(1./PSI(I))
      R(I)=RZ/ALET**2.355
1     CONTINUE
      RETURN
      END
```

```
C       AUHIGH.FOR
C       HEIGHT IN MM AND WATER MM FOR PARAMETER FILE.
        TYPE 20
20      FORMAT(' STACK DATA FILE NAME'/)
        CALL ASSIGN(3,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        READ(3)NL
        Z=0.
        ZL=0.
        ZW=0.
        ZLW=0.
        DO 21 I=1,NL
        READ(3)H,F
        H=.5*H
        HW=H*F
        Z=Z+H+ZL
        ZW=ZW+HW+ZLW
        WRITE(2,1)I,Z,ZW
1       FORMAT(1X,I3,2F8.4)
        ZLW=HW
21      ZL=H
        END
```

```
C       AUITH.FOR
C       STORING ITTERATION VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 5
5       FORMAT(' ERROR LIMIT')
        ACCEPT 6,EL
6       FORMAT(F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        IF(SZ.LT.EL)WRITE(2)FLOAT(IT)
        GO TO 3
4       END
```

```
C       AUNH.FOR
C       WRITING THE NUMBER OF TRACKS PER PARTICLE FROM A TRAJECTORY
C       FILE TO A HISTOGRAM FILE.
        DIMENSION P(6),LS(100),AS(100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
3       READ(1,END=4)P,M,(LS(I),AS(I),I=1,M)
        WRITE(2)FLOAT(M)
        GO TO 3
4       END
```

```
C       AUPAR.FOR
C       GENERATION OF THE LAYER PARAMETER FILE
        DIMENSION P(7,10),LT(100),PL(7)
        TYPE 3
3       FORMAT(' PARAMETER FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 1
1       FORMAT(' NUMBER OF DIFFERENT LAYER TYPES (MAX. = 10)')
        ACCEPT 2,NT
2       FORMAT(7I10)
        NL=0
        TYPE 4
4       FORMAT(' H0, F(H2O), RHO, CLET, PSI, V0, HW')
        DO 5 I=1,NT
5       ACCEPT 6,(P(J,I),J=1,7)
6       FORMAT(7F10.0)
12      N=0
        TYPE 7
7       FORMAT(' LAYER TYPE IN SEQUENCE (0 FOR EXIT)')
8       N=N+1
        ACCEPT 2,LT(N)
        IF(LT(N).NE.0)GO TO 8
        N=N-1
        IF(N.EQ.0)GO TO 11
        TYPE 9
9       FORMAT(' NUMBER OF SEQUENCES')
        ACCEPT 2,NS
        DO 10 I=1,NS
        DO 10 J=1,N
        NL=NL+1
10      WRITE(1)(P(K,LT(J)),K=1,7)
        GO TO 12
11      REWIND 1
        WRITE(2)NL
        DO 13 I=1,NL
        READ(1)PL
13      WRITE(2)PL
        END
```

```
C       AUPARC.FOR
C       CHANGING CLET, PSI, V0, AND HW
        DIMENSION P(7)
        TYPE 3
3       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        TYPE 1
1       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 4
4       FORMAT(' OLD VALUES OF CLET,PSI,V0,HW')
        ACCEPT 5,CLETO,PSIO,V0O,HWO
        TYPE 2
2       FORMAT(' NEW VALUES OF CLET,PSI,V0,HW')
        ACCEPT 5,CLET,PSI,V0,HW
5       FORMAT(7F10.0)
        READ(1)NL
        WRITE(2)NL
        DO 10 N=1,NL
        READ(1)P
        IF(CLETO.NE.P(4).OR.PSIO.NE.P(5).OR.V0O.NE.P(6).OR.HWO.NE.P(7))
        *GO TO 6
        WRITE(2)(P(J),J=1,3),CLET,PSI,V0,HW
        GO TO 10
6       WRITE(2)P
10      CONTINUE
11      END
```

```
C       AUPARI.FOR
C       GENERATION OF THE LAYER PARAMETER FILE
C       WITH INDIVIDUAL LAYER THICKNESSES.
        DIMENSION P(7,10),LT(100),PL(7)
        TYPE 3
3       FORMAT(' PARAMETER FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 1
1       FORMAT(' NUMBER OF DIFFERENT LAYER TYPES (MAX. = 10)')
        ACCEPT 2,NT
2       FORMAT(7I10)
        NL=0
        TYPE 4
4       FORMAT(' H0, F(H2O), RH0, CLET, PSI, V0, HW')
        DO 5 I=1,NT
5       ACCEPT 6,(P(J,I),J=1,7)
6       FORMAT(7F10.0)
12      N=0
        TYPE 7
7       FORMAT(' LAYER TYPE IN SEQUENCE (0 FOR EXIT)')
8       N=N+1
        ACCEPT 2,LT(N)
        IF(LT(N).NE.0)GO TO 8
        N=N-1
        IF(N.EQ.0)GO TO 11
        TYPE 9
9       FORMAT(' NUMBER OF SEQUENCES')
        ACCEPT 2,NS
        DO 10 I=1,NS
        DO 10 J=1,N
        NL=NL+1
        TYPE 14
14      FORMAT(' H0,H1')
        ACCEPT 6,H0,H1
        IF(H0.LE.0.)GO TO 15
        L=LT(J)
        WRITE(1)H0,P(2,L),(H0-H1)/H0,(P(K,L),K=4,7)
        GO TO 10
15      WRITE(1)(P(K,LT(J)),K=1,7)
10      CONTINUE
        GO TO 12
11      REWIND 1
        WRITE(2)NL
        DO 13 I=1,NL
        READ(1)PL
13      WRITE(2)PL
        END
```

```
C       AUPARP.FOR
C       PRINTING THE LAYER PARAMETERS.
        DIMENSION P(7)
        TYPE 3
3       FORMAT(' PARAMETER FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'NEW')
        TYPE 4
4       FORMAT(' N, H0, F(H20), RH0, CLET, PSI, V0, HW')
        READ(1)NL
        DO 10 N=1,NL
        READ(1)P
        TYPE 12,N,P
12      FORMAT(1X,I5,7F9.4)
10      CONTINUE
11      END
```

```
C       AURCOM.FOR
C       COMBINING RAW AU DATA FILES INTO ONE FILE.
        TYPE 1
1       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
2       TYPE 4
4       FORMAT(' LAYER RANGE')
        ACCEPT 5,L1,L2
        IF(L1.EQ.0.AND.L2.EQ.0)GO TO 8
5       FORMAT(2I10)
        TYPE 3
3       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
6       READ(1,END=7)X,Y,L,A
D       IF(L.LT.111)L=L+100
        IF(L.LT.L1)GO TO 6
        IF(L.GT.L2)GO TO 7
        WRITE(2)X,Y,L,A
        GO TO 6
7       CALL CLOSE(1)
        GO TO 2
8       END
```

```
C       AUREGP.FOR
C       PLOT OF RREG(DELTA;Z) IN CN, LEXAN, AND CR-39.
        DIMENSION S(4),Y(90,3),REG(3)
        COMMON R1(3),ALAM(3),NL(3),FR(3),H(3),F(3),RR,RMU
        DATA R1/.004699,5.686E-5,.01216/,ALAM/1.036,.894,2.899/,
     *F/1.302,1.117,1.27/,RMU/3.768/,PI/3.1415927/
1       TYPE 2
2       FORMAT(' FRACTION OF LAYER REMOVED IN CN, LEXAN, CR-39')
        ACCEPT 3,FR
3       FORMAT(7F10.0)
4       TYPE 5
5       FORMAT(' Z')
        ACCEPT 3,Z
        DPR=180./PI
        DO 6 I=2,90,2
        X=I
        DELTA=X/DPR
        CALL RREG(Z,DELTA,REG)
        DO 6 J=1,3
6       Y(I,J)=REG(J)
        CALL RREG(Z,PI/2.,REG)
        DO 7 J=1,3
        S(1)=0.
        S(2)=89.
        S(3)=0.
        S(4)=REG(J)
        CALL SCALE(S)
        DO 8 K=4,90,2
8       CALL VECTOR(S,FLOAT(K-2),Y(K-2,J),FLOAT(K),Y(K,J))
        ACCEPT 3,XL
7       CONTINUE
        GO TO 4
        END
C
C
        SUBROUTINE RREG(Z,DELTA,R)
C
        DIMENSION R(3)
        COMMON R1(3),ALAM(3),NL(3),FR(3),H(3),F(3),RR,RMU
        DO 1 I=1,3
        CALL ETA(FR(I),DELTA,0.,A,P)
        R(I)=0.
        IF(A.LE.0.)GO TO 1
        RZ=R1(I)*Z**RMU
        CD=COS(DELTA)
        SD=SIN(DELTA)
        X=(1.-FR(I))*(1.-FR(I))
        T=ATAN2((FR(I)*FR(I)-X*CD*CD)*SD,(1.-X*SD*SD)*CD)
        R(I)=.5*RZ*(T/SQRT(1.+T*T))**(1./ALAM(I))
        DR=R(I)/2.
4       ST=(R(I)/RZ)**ALAM(I)
        CALL ETA(FR(I),DELTA,ATAN2(ST,SQRT(1.-ST*ST)),A,P)
        IF(A.LE.0.)GO TO 2
        R(I)=R(I)+DR
        GO TO 3
2       R(I)=R(I)-DR
3       DR=DR*.5
        IF(DR.GT..001)GO TO 4
1       CONTINUE
        RETURN
        END
```

```
C       AURLVX.FOR
C       PLOTTING RAW AUTOMATED HZE LAYER VS. X ON THE TEKTRONIX
C       AS LINE SEGMENTS ORIENTED LINEARLY WITH Y
        DIMENSION S(4)
        TYPE 1
1       FORMAT(' ABSISSA = LAYER, ORDINATE = X, ANGLE OF LINE = Y'
     *,//,' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' X RANGE, Y RANGE, LAYER RANGE')
        ACCEPT 3,S(3),S(4),Y1,Y2,L1,L2
3       FORMAT(4F10.0,2I10)
        S(4)=S(4)-.0001
        S(1)=L1
        S(2)=L2-.0001
        CALL SCALE(S)
        RH=.005*(S(2)-S(1))
        RV=.005*(S(4)-S(3))
4       READ(1,END=5)X,Y,IZ,A
        IF(IZ.LT.L1)GO TO 4
        IF(IZ.GT.L2)GO TO 5
        IF(X.LT.S(3).OR.X.GT.S(4))GO TO 4
        IF(Y.LT.Y1.OR.Y.GT.Y2)GO TO 4
        Z=IZ
        T=3.1415927*(Y-Y1)/(Y2-Y1)
        H=RH*COS(T)
        V=RV*SIN(T)
        CALL VECTOR(S,Z-H,X-V,Z+H,X+V)
        GO TO 4
5       END
```

```
C       AURMER.FOR
C       COMBINING RAW DATA FILES BY MERGING TWO FILES TYPICALLY FROM
C       DIFFERENT MATERIALS.
        TYPE 1
1       FORMAT(' INPUT FILE #1'/)
        CALL ASSIGN(1,'A',-1)
        TYPE 2
2       FORMAT(' INPUT FILE #2'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 3
3       FORMAT(' OUTPUT FILE'/)
        CALL ASSIGN(3,'A',-1)
        READ(1)X1,Y1,L1,A1
        READ(2)X2,Y2,L2,A2
6       IF(L1.GT.L2)GO TO 4
        WRITE(3)X1,Y1,L1,A1
        READ(1,END=5)X1,Y1,L1,A1
        GO TO 6
4       WRITE(3)X2,Y2,L2,A2
        READ(2,END=7)X2,Y2,L2,A2
        GO TO 6
5       WRITE(3)X2,Y2,L2,A2
        READ(2,END=8)X2,Y2,L2,A2
        GO TO 5
7       WRITE(3)X1,Y1,L1,A1
        READ(1,END=8)X1,Y1,L1,A1
        GO TO 7
8       END
```

```
C       AURPR.FOR
C       PRINTING THE OUTPUT OF AUSTO.
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 7
7       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 5
5       FORMAT(' X RANGE, Y RANGE, LAYER RANGE')
        ACCEPT 6,X1,X2,Y1,Y2,L1,L2
6       FORMAT(4F10.0,2I10)
3       READ(1,END=2)X,Y,IZ,A
        IF(X.LT.X1.OR.X.GT.X2.OR.Y.LT.Y1.OR.Y.GT.Y2.OR.IZ.LT.L1)GO TO 3
        IF(IZ.GT.L2)GO TO 2
        WRITE(2,4)X,Y,IZ,A
4       FORMAT(1X,2F8.3,I4,G11.4)
        GO TO 3
2       END
```

```
C       AURTLP.FOR
C       PLOTTING AUTOMATED HZE DATA POINTS ON THE TEKTRONIX
C       AS A LINE LINEARLY ORIENTED WITH LAYER NO.
        DIMENSION S(4)
        TYPE 1
1       FORMAT(' ABSISSA = X, ORDINATE = Y, ANGLE OF LINE = Z'
     *,//,' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' X RANGE, Y RANGE, Z RANGE')
        ACCEPT 3,S,Z1,Z2
3       FORMAT(6F10.0)
        CALL SCALE(S)
        RH=.005*(S(2)-S(1))
        RV=.005*(S(4)-S(3))
4       READ(1,END=5)X,Y,IZ,A
        Z=IZ
        IF(X.LT.S(1).OR.Y.LT.S(3))GO TO 4
        IF(Z.LT.Z1.OR.Z.GT.Z2)GO TO 4
        T=3.1415927*(Z-Z1)/(Z2-Z1)
        H=RH*COS(T)
        V=RV*SIN(T)
        IF(X+H.GT.S(2).OR.Y+V.GT.S(4))GO TO 4
        CALL VECTOR(S,X-H,Y-V,X+H,Y+V)
        GO TO 4
5       END
```

```
C       AURTPP.FOR
C       PLOTTING RAW AUTOMATED HZE DATA AS POINTS ON THE TEKTRONIX.
        DIMENSION S(4)
        TYPE 1
1       FORMAT(' ABSISSA = X, ORDINATE = Y'
     *,//,' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' X RANGE, Y RANGE, LAYER RANGE')
        ACCEPT 3,S,L1,L2
        Z1=L1
        Z2=L2
3       FORMAT(4F10.0,2I10)
        CALL SCALE(S)
        RH=.005*(S(2)-S(1))
        RV=.005*(S(4)-S(3))
4       READ(1,END=5)X,Y,IZ,A
        IF(IZ.LT.L1)GO TO 4
        IF(IZ.GT.L2)GO TO 5
        IF(X.LT.S(1).OR.Y.LT.S(3).OR.X.GT.S(2).OR.Y.GT.S(4))GO TO 4
        CALL BPOINT(S,X,Y)
        GO TO 4
5       ACCEPT 6,I
6       FORMAT(2A1)
        END
```

```
C       AURTTP.FOR
C       PLOTTING RAW AUTOMATED HZE DATA POINTS AND TRACED
C       TRAJECTORIES ON THE TEKTRONIX.
        DIMENSION S(4),ZA(1000),LA(100),AA(100)
        DPR=45./ATAN(1.)
        TYPE 1
1       FORMAT(' RAW DATA FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 6
6       FORMAT(' TRACED DATA FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'RDO')
        TYPE 11
11      FORMAT(' PARAMETER FILE NAME'/)
        CALL ASSIGN(3,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' X RANGE, Y RANGE, LAYER RANGE')
        ACCEPT 3,S,IZ1,IZ2
3       FORMAT(4F10.0,2I10)
        CALL SCALE(S)
        RH=.005*(S(2)-S(1))
        RV=.005*(S(4)-S(3))
        Z1=IZ1
        Z2=IZ2
4       READ(1,END=5)X,Y,IZ,A
        IF(IZ.LT.IZ1)GO TO 4
        IF(IZ.GT.IZ2)GO TO 5
        IF(X.LT.S(1).OR.Y.LT.S(3).OR.X.GT.S(2).OR.Y.GT.S(4))GO TO 4
        CALL BPOINT(S,X,Y)
        GO TO 4
5       READ(3)NL
        READ(3)H
        H=H/2.
        HL=H
        ZA(1)=H
        DO 7 I=2,NL
        READ(3)H
        H=.5*H
        ZA(I)=ZA(I-1)+H+HL
7       HL=H
8       READ(2,END=9)ALPHA,DELTA,SDEL,X0,Y0,SIG,M,(LA(I),AA(I),I=1,M)
        IF(M.LT.3)GO TO 8
        IF(LA(1).LT.IZ1)GO TO 8
        IF(LA(M).GT.IZ2)GO TO 9
        DD=COS(DELTA)/SIN(DELTA)
        DX=DD*COS(ALPHA)
        DY=DD*SIN(ALPHA)
        Z=ZA(LA(1))
        XB=X0+DX*Z
        YB=Y0+DY*Z
        IF(XB.LT.S(1).OR.XB.GT.S(2).OR.YB.LT.S(3).OR.YB.GT.S(4))GO TO 8
        Z=ZA(LA(M))
        XT=X0+DX*Z
        YT=Y0+DY*Z
        IF(XT.LT.S(1).OR.XT.GT.S(2).OR.YT.LT.S(3).OR.YT.GT.S(4))GO TO 8
        CALL VECTOR(S,XB,YB,XT,YT)
        GO TO 8
9       ACCEPT 10,I
10      FORMAT(2A1)
        END
```

```
C       AURVLP.FOR
C       PLOTTING THE RAW AUTOMATED DATA ON THE VERSATEC AS A LINE
C       ORIENTED AT AN ANGLE WHICH IS LINEAR IN THE LAYER NUMBER.
        CALL PLOTS(0,0,0)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' X RANGE, Y RANGE, Z RANGE')
        ACCEPT 3,X1,X2,Y1,Y2,Z1,Z2
3       FORMAT(7F10.0)
        TYPE 4
4       FORMAT(' AREA SCALE SIZE')
        ACCEPT 3,AM
        CALL AXIS(.55,.55,'X',-1,10.,0.,X1,.1*(X2-X1))
        CALL AXIS(.55,.55,'Y',1,10.,90.,Y1,.1*(Y2-Y1))
5       READ(1,END=6)X,Y,L,A
        IF(X.LT.X1.OR.X.GT.X2.OR.Y.LT.Y1.OR.Y.GT.Y2)GO TO 5
        Z=L
        IF(Z.LT.Z1.OR.Z.GT.Z2)GO TO 5
        H=.1
        IF(AM.NE.0.)H=.1*A/AM
        ANG=180.*(Z-Z1)/(Z2-Z1)
        CALL SYMBOL(.55+10.*(X-X1)/(X2-X1),.55+10.*(Y-Y1)/(Y2-Y1),
     *H,13,ANG,-1)
        GO TO 5
6       CALL PLOT(0.,0.,999)
        END
```

```
C       AUSAH.FOR
C       STORING SIGMA OF AREA VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 5
5       FORMAT(' ERROR LIMIT')
        ACCEPT 6,EL
6       FORMAT(F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        IF(SZ.LT.EL)WRITE(2)RMS
        GO TO 3
15      FORMAT(/1X,'ALPHA=',F6.1,', DELTA=',F4.1,'+/-',F4.1,', X0=',F5.2,
     *', Y0=',F5.2/1X,'SIG.POS.=',F4.2,', ITT.=',I2,', NUM.PTS.=',I2
     */1X,'Z=',F4.1,'+/-',F3.1,', ST.PT.=',
     *F5.2,'+/-',F5.2,' CORR.=',F6.3,' RMS.DEV.=',F6.4)
4       END
```

```
C       AUSDH.FOR
C       STORING SIGMA DELTA VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
6       FORMAT(F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        WRITE(2)57.3*SD
        GO TO 3
4       END
```

```
C       AUSDH.FOR
C       STORING SIGMA DELTA VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
6       FORMAT(F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        WRITE(2)57.3*SD
        GO TO 3
4       END
```

```
C       AUSEL.FOR
C       SELECTING THE TRACED TRAJECTORIES WITH GREATER THAN A MINIMUM
C       NUMBER OF TRACKS.
        DIMENSION P(6),L(85),A(85)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 5
5       FORMAT(' MINIMUM NUMBER OF TRACKS PER TRAJECTORY')
        ACCEPT 6,MM
6       FORMAT(I10)
3       READ(1,END=4)P,M,(L(I),A(I),I=1,M)
        IF(M.GE.MM)WRITE(2)P,M,(L(I),A(I),I=1,M)
        GO TO 3
4       END
```

```
C       AUSELA.FOR
C       SELECTING THE TRACED TRAJECTORIES WITH PARTICULAR VALUES
C       OF ALPHA AND DELTA
        DIMENSION P(6),L(85),A(85)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
7       TYPE 5
5       FORMAT(' ALPHA, DELTA (DEGREES)')
        ACCEPT 6,ALPHA,DELTA
6       FORMAT(2F10.0)
        IF(ALPHA.EQ.0..AND.DELTA.EQ..0)GO TO 4
3       READ(1,END=4)P,M,(L(I),A(I),I=1,M)
        IF(ABS(ALPHA-57.3*P(1)).GT..15.OR.ABS(DELTA-57.3*P(2)).GT..15)
     *GO TO 3
        WRITE(2)P,M,(L(I),A(I),I=1,M)
        GO TO 7
4       END
```

```
C       AUSH.FOR
C       WRITING THE SIGMA FROM A TRAJECTORY
C       FILE TO A HISTOGRAM FILE FOR MORE THAN 3 TRACKS/TRAJECTORY.
        DIMENSION P(6),LS(100),AS(100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
3       READ(1,END=4)P,M,(LS(I),AS(I),I=1,M)
        IF(M.GE.3)WRITE(2)P(6)
        GO TO 3
4       END
```

```
C       AUSTOE.FOR
C       STORING POSITIONS AND AREAS OF TRACKS FOR TRACING.
C       EYECOM VERSION.
C
C       INITIALIZING.
C
10      FORMAT(7F10.0)
15      FORMAT(7I10)
        LOGICAL*1 LS(15,20)
        DIMENSION XP(200),YP(200)
        DIMENSION IA(45),AREA(200)
        CALL IPOKE("164202,"13)
        CALL TRANI(IA)
        IZ=1
        IFLAG=0
        IZL=0
        WRITE(2)0
3       TYPE 2
2       FORMAT(' STAGE TRANSLATION DX, DY (MM) (0,0 FOR CONTINUE)')
        ACCEPT 10,DX,DY
        IF(DX.EQ.0..AND.DY.EQ.0.)GO TO 4
        CALL TRAN(IA,DX,DY)
        GO TO 3
C
C       INPUT OF PARAMETERS FROM CONSOLE.
C
4       TYPE 40
40      FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'NEW')
22      TYPE 60
60      FORMAT(' APPROXIMATE SCALE FACTOR (PP/MM)')
70      ACCEPT 10,S
        DO 14 I=1,15
        DO 14 J=1,20
14      LS(I,J)=0
        TYPE 12
12      FORMAT(' REMOVE SAMPLE. SLIGHTLY DEFOCUS. THEN INPUT THE'/
        *' RELATIVE WEIGHT OF A CLEAR FIELD FOR THE GRAY LEVEL CUTOFF.
        *'/' (DEFAULT=0.5).')
        ACCEPT 10,GWT
        IF(GWT.EQ.0.)GWT=.5
        CALL SHADE(LS,GWT)
        CALL IPOKE("164202,"13)
        TYPE 13
13      FORMAT(' PLACE THE FIELD ON A CLEAR REGION OF THE SAMPLE,'/
        *' SLIGHTLY DEFOCUSED. PRESS CR')
        ACCEPT 10,A
        GWT=1.-GWT
        CALL SHADE(LS,GWT)
        TYPE 5
5       FORMAT(' POSITION A FEATURE ON THE LOWER LEFT CORNER,'/
        *' TRANSMIT A CARRIAGE RETURN'/
        *' PLACE CURSOR ON THE FEATURE, AND TRANSMIT THE CURSOR'/
        *' POSITION WITH AN "ENQ" OR CNTRL E')
        CALL IPOKE("164202,"13)
        ACCEPT 10,DUM
        CALL TRAN(IA,-128./S,-96./S)
        CALL IPOKE("164202,"12)
        CALL CURSOR(IX,IY)
        CALL IPOKE("164202,"13)
        DX=-.01*AINT(38400./S)
        DY=-.01*AINT(28800./S)
        CALL TRAN(IA,DX,DY)
        TYPE 6
```

```
6       FORMAT(' ALLIGN CURSOR WITH THE FEATURE AND AGAIN'/
        *' TRANSMIT ITS POSTION.')
        CALL CURSOR(IXP,IYP)
        S1=(DX*DX+DY*DY)/(FLOAT(IXP-IX)**2+FLOAT(IYP-IY)**2)
        S=SQRT(S1)
        S1=4.*S1
        TYPE 7
7       FORMAT(' COORDINATES OF LOWER LEFT CORNER OF REGION TO BE
        * SCANNED'/' ON SAMPLE (MM,MM.')
        ACCEPT 10,X1,Y1
        TYPE 8
8       FORMAT(' COORDINATES OF UPPER RIGHT CORNER OF SCAN REGION(MM,MM')
        ACCEPT 10,X2,Y2
        TYPE 9
9       FORMAT(' X POSITION OF FIDUCIAL FOR SAMPLE ANGLE MEASUREMENT
        * (MM).')
        ACCEPT 10,XA
        TYPE 11
11      FORMAT(' WIDTH OF MASK AROUND EDGE OF FIELD (PP)')
        ACCEPT 15,IMASK
        TYPE 185
185     FORMAT(' RANGE OF ACCEPTABLE FEATURE AREAS(PP)')
        ACCEPT 10,AR1,AR2
C
C       BEGIN MEASUREMENT OF NEW LAYER.
C
190     CALL IPOKE("164202,"13)
        ENDFILE 2
        TYPE 191
191     FORMAT(' LAYER NUMBER (0 FOR EXIT, -1 FOR SET UP)')
        ACCEPT 15,IZ
        IF(IZ.EQ.IZL.OR.IFLAG.EQ.0)GO TO 25
        REWIND 2
26      READ(2,END=25)X,Y,L,A
        WRITE(1)X,Y,L,A
        GO TO 26
25      REWIND 2
        IZL=IZ
        IFLAG=1
        IF(IZ)22,280,192
192     TYPE 16
16      FORMAT(' CENTER SAMPLE ORIGIN IN FIELD, PLACE CURSOR'/
        *' ON SAMPLE ORIGIN, AND TRANSMIT CURSOR COORDINATES
        *'/' WITH "ENQ OR CNTRL E"')
        CALL CURSOR(IX,IY)
        XF=-IX*S
        YF=IY*S
        X=XA-320.*S
        Y=240.*S
        CALL TRAN(IA,X-XF,Y-YF)
        XF=X
        YF=Y
        TYPE 17
17      FORMAT(' PLACE CURSOR ON FIDUCIAL.  TRANSMIT CURSOR
        *COORDINATES')
        CALL CURSOR(IX,IY)
        TP=(YF-S*IY)/(XF+S*IX)
        CP=1./SQRT(1.+TP*TP)
        SP=TP*CP
        XMIN=AMIN1(X1*CP-Y1*SP,X1*CP-Y2*SP)-S*IMASK
        XMAX=AMAX1(X2*CP-Y1*SP,X2*CP-Y2*SP)-IMASK*S
        YMIN=AMIN1(X1*SP+Y1*CP,X2*SP+Y1*CP)+(480.-IMASK)*S
        YMAX=AMAX1(X1*SP+Y2*CP,X2*SP+Y2*CP)+S*(480.-IMASK)
        DX=S*(640.-2.*IMASK)
```

```
      DY=S*(480.-2.*IMASK)
      ADX=DX
      ADY=0.
      CALL TRAN(IA,XMIN-XF,YMIN-YF)
      TIME=SECNDS(0.)
      XF=XMIN
      YF=YMIN
      XR=320.-IMASK
      YR=240.-IMASK
C
C     MEASURE LAYER
C
1     IF(SECNDS(TIME).LT..5)GO TO 1
      CALL DIG
      CALL IPOKE("164202,"13)
      CALL TRAN(IA,ADX,ADY)
      TIME=SECNDS(6.)
      CALL IPOKE("164202,"43)
      CALL FEATUR(LS,N,XP,YP,AREA)
      IF(N.EQ.0)GO TO 23
      DO 18 I=1,N
      IF(ABS(XP(I)-320.).GT.XR.OR.ABS(YP(I)-240.).GT.YR.OR
     *.AREA(I).LT.AR1.OR.AREA(I).GT.AR2)GO TO 18
      XS=XF+S*XP(I)
      YS=YF-S*YP(I)
      X=XS*CP+YS*SP
      Y=-XS*SP+YS*CP
      IF(X.LT.X1.OR.X.GT.X2.OR.Y.LT.Y1.OR.Y.GT.Y2)GO TO 18
21    FORMAT(' ',72A1)
      CALL IPOKE("164204,INT(XP(I)))
      CALL IPOKE("164206,INT(YP(I)))
      CALL IPOKE("164236,0)
      WRITE(2)X,Y,IZ,S1*AREA(I)
18    CONTINUE
23    XF=XF+ADX
      YF=YF+ADY
      X=XF+ADX
      CALL IPOKE("164202,"13)
      IF(X.GE.XMIN-.01.AND.X.LT.XMAX.AND.ADX.NE.0.)GO TO 1
      IF(ADX.EQ.0.)GO TO 20
      IF(YF+DY.GE.YMAX)GO TO 19
      ADY=DY
      ADX=0.
      DX=-DX
      GO TO 1
20    ADX=DX
      ADY=0.
      GO TO 1
19    CALL TRAN(IA,-320.*S-XF,240.*S-YF)
      DO 24 I=1,10
      TYPE 21,7
      DO 24 J=1,5000
24    CONTINUE
      GO TO 190
280   END
C
C
      SUBROUTINE SHADE(LS,GWT)
C     ADDING GWT*AVERAGE GRAY VALUES TO THE SHADING ARRAY LS.
      LOGICAL*1 LS(15,20)
      CALL DIG
      IXA=0
      DO 1 I=1,20
      IYA=0
```

```
      DO 2 J=1,15
      ISUM=LS(J,I)
      IF(ISUM.LT.0) ISUM=ISUM+256
      CALL ASUM(IXA,IYA,IADD)
      ISUM=ISUM+GWT*IADD/32.
      IF(ISUM.GT.127) ISUM=ISUM-256
      LS(J,I)=ISUM
2     IYA=IYA+32
1     IXA=IXA+32
      RETURN
      END
```

```
C       AUSTOF.FOR
C       STORING POSITIONS AND AREAS OF TRACKS FOR TRACING.
C       EYECOM VERSION.
C       MEASURES EACH EVENT FROM INPUT FILE INDIVIDUALLY.
C
C       INITIALIZING.
C
10      FORMAT(7F10.0)
15      FORMAT(7I10)
        LOGICAL*1 LS(15,20)
        DIMENSION XP(200),YP(200),XI(2000),YI(2000)
        DIMENSION IA(45),AREA(200)
        CALL IPOKE("164202,"13)
        CALL TRANI(IA)
        IZ=1
        IFLAG=0
        IZL=0
        LEND=1
        WRITE(2)0
3       TYPE 2
2       FORMAT(' STAGE TRANSLATION DX, DY (MM (0,0 FOR CONTINUE)')
        ACCEPT 10,DX,DY
        IF(DX.EQ.0..AND.DY.EQ.0.)GO TO 4
        CALL TRAN(IA,DX,DY)
        GO TO 3
C
C       INPUT OF PARAMETERS FROM CONSOLE.
C
4       TYPE 27
27      FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(3,'A',-1)
        TYPE 40
40      FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'NEW')
22      TYPE 60
60      FORMAT(' APPROXIMATE SCALE FACTOR (PP/MM)')
70      ACCEPT 10,S
        DO 14 I=1,15
        DO 14 J=1,20
14      LS(I,J)=0
        TYPE 12
12      FORMAT(' REMOVE SAMPLE. SLIGHTLY DEFOCUS. THEN INPUT THE'/
        *' RELATIVE WEIGHT OF A CLEAR FIELD FOR THE GRAY LEVEL CUTOFF.
        *'/' (DEFAULT=0.5).')
        ACCEPT 10,GWT
        IF(GWT.EQ.0.)GWT=.5
        CALL SHADE(LS,GWT)
        CALL IPOKE("164202,"13)
        TYPE 13
13      FORMAT(' PLACE THE FIELD ON A CLEAR REGION OF THE SAMPLE,'/
        *' SLIGHTLY DEFOCUSED. PRESS CR')
        ACCEPT 10,A
        GWT=1.-GWT
        CALL SHADE(LS,GWT)
        TYPE 5
5       FORMAT(' POSITION A FEATURE ON THE LOWER LEFT CORNER,'/
        *' TRANSMIT A CARRIAGE RETURN'/
        *' PLACE CURSOR ON THE FEATURE, AND TRANSMIT THE CURSOR'/
        *' POSITION WITH AN "ENQ" OR CNTRL E')
        CALL IPOKE("164202,"13)
        ACCEPT 10,DUM
        CALL TRAN(IA,-128./S,-96./S)
        CALL IPOKE("164202,"12)
        CALL CURSOR(IX,IY)
```

```
      CALL IPOKE("164202,"13)
      DX=-.01*AINT(38400./S)
      DY=-.01*AINT(28800./S)
      CALL TRAN(IA,DX,DY)
      TYPE 6
6     FORMAT(' ALLIGN CURSOR WITH THE FEATURE AND AGAIN'/
     *' TRANSMIT ITS POSTION.')
      CALL CURSOR(IXP,IYP)
      S1=(DX*DX+DY*DY)/(FLOAT(IXP-IX)**2+FLOAT(IYP-IY)**2)
      S=SQRT(S1)
      S1=4.*S1
      TYPE 9
9     FORMAT(' X POSITION OF FIDUCIAL FOR SAMPLE ANGLE MEASUREMENT
     * (MM).')
      ACCEPT 10,XA
      TYPE 185
185   FORMAT(' RANGE OF ACCEPTABLE FEATURE AREAS(PP)')
      ACCEPT 10,AR1,AR2
C
C     BEGIN MEASUREMENT OF NEW LAYER.
C
190   IF(LEND.EQ.0)GO TO 28
      CALL IPOKE("164202,"13)
      ENDFILE 2
      TYPE 191
191   FORMAT(' LAYER NUMBER (0 FOR EXIT, -1 FOR SET UP, -2 FOR RE-DO)')
      ACCEPT 15,IZ
      IRDO=0
      IF(IZ.EQ.IZL.OR.IFLAG.EQ.0.OR.IZ.EQ.-2)GO TO 25
      REWIND 2
26    READ(2,END=25)X,Y,L,A
      WRITE(1)X,Y,L,A
      GO TO 26
25    REWIND 2
      IFLAG=1
      IF(IZ.NE.-2)GO TO 48
      IZ=IZL
      IRDO=1
      GO TO 192
48    IZL=IZ
      IF(IZ)22,280,192
192   TYPE 16
16    FORMAT(' CENTER SAMPLE ORIGIN IN FIELD, PLACE CURSOR'/
     *' ON SAMPLE ORIGIN, AND TRANSMIT CURSOR COORDINATES
     *'/' WITH "ENQ OR CNTRL E"')
      CALL CURSOR(IX,IY)
      XF=-IX*S
      YF=IY*S
      X=XA-320.*S
      Y=240.*S
      CALL TRAN(IA,X-XF,Y-YF)
      XF=X
      YF=Y
      TYPE 17
17    FORMAT(' PLACE CURSOR ON FIDUCIAL.  TRANSMIT CURSOR
     *COORDINATES')
      CALL CURSOR(IX,IY)
      TP=(YF-S*IY)/(XF+S*IX)
      CP=1./SQRT(1.+TP*TP)
      SP=TP*CP
      IDIR=1
      LEND=0
      IF(IRDO.EQ.1)GO TO 49
C
```

```
C       READ EXPECTED TRACK POSITIONS
C
28      NEV=0
        XI1=1.E37
        XI2=-XI1
        YI1=XI1
        YI2=XI2
30      READ(3,END=29)X,Y,L,A
        IF(L-IZ)30,31,32
31      NEV=NEV+1
        XI1=AMIN1(XI1,X)
        XI2=AMAX1(XI2,X)
        YI1=AMIN1(YI1,Y)
        YI2=AMAX1(YI2,Y)
        IF(NEV.GT.1)GO TO 34
        XI(1)=X
        YI(1)=Y
        GO TO 35
34      DO 33 I=NEV,2,-1
        IF(X.GT.XI(I-1))GO TO 36
        XI(I)=XI(I-1)
33      YI(I)=YI(I-1)
        I=1
36      XI(I)=X
        YI(I)=Y
35      IF(NEV-2000)30,37,37
32      BACKSPACE 3
29      LEND=1
37      NINT=1
        YI2=YI2+.001
        IF(NEV-1)190,38,39
39      W=YI2-YI1
        FLU=(NEV-1)/(W*(XI2-XI1))
        NINT=AMAX1(1.,.5+W*SQRT(FLU/3.))
38      YINT=W/NINT
C
C       MEASURE LAYER
C
49      TYPE 47,(XI(K),YI(K),K=1,NEV)
47      FORMAT(1X,2F10.3)
        IFNF=0              !NEW FIELD FLAG
        IFNP=0              !NEW PICTURE FLAG
        ASSIGN 20 TO MV
        DO 1 INTE=1,NINT
        Y2=INTE*YINT+YI1
        Y1=Y2-YINT
        IF(IDIR)42,280,41
41      I1=1
        I2=NEV
        GO TO 43
42      I1=NEV
        I2=1
43      DO 44 I=I1,I2,IDIR
        IF(YI(I).LT.Y1.OR.YI(I).GE.Y2)GO TO 44
        XL=XI(I)*CP-YI(I)*SP-320.*S
        YL=XI(I)*SP+YI(I)*CP+240.*S
        CALL IPOKE("164202,"12)
        CALL TRAN(IA,XL-XF,YL-YF)
        TIME=SECNDS(0.)
        IFNF=1
45      IF(IFNP.EQ.0)GO TO 23
        CALL IPOKE("164202,"43)
        CALL FEATUR(LS,N,XP,YP,AREA)
        IF(N.EQ.0.OR.N.GE.200)GO TO 23
```

```
      DM=1.E37
      DO 18 J=1,N
      DX=XP(J)-320.
      DY=YP(J)-240.
      DS=DX*DX+DY*DY
      IF(DS.GT.DM)GO TO 18
      DM=DS
      IM=J
18    CONTINUE
      XS=XF+S*XP(IM)
      YS=YF-S*YP(IM)
      X=XS*CP+YS*SP
      Y=-XS*SP+YS*CP
21    FORMAT(' ',72A1)
      CALL IPOKE("164204,INT(XP(IM))
      CALL IPOKE("164206,INT(YP(IM))
      CALL IPOKE("164236,0)
      WRITE(2)X,Y,IZ,S1*AREA(IM
      IFNP=0
23    XF=XL
      YF=YL
      GO TO MV
20    IF(IFNP.EQ.1.OR.IFNF.EQ.0)GO TO 44
46    IF(SECNDS(TIME).LT..5)GO TO 46
      CALL DIG
      CALL IPOKE("164202,"13)
      IFNP=1
      IFNF=0
44    CONTINUE
1     IDIR=-IDIR
      ASSIGN 19 TO MV
      GO TO 45
19    CALL TRAN(IA,-320.*S-XF,240.*S-YF)
      DO 24 I=1,10
      TYPE 21,7
      DO 24 J=1,5000
24    CONTINUE
      GO TO 190
280   END
C
C
      SUBROUTINE SHADE(LS,GWT)
C     ADDING GWT*AVERAGE GRAY VALUES TO THE SHADING ARRAY LS.
      LOGICAL*1 LS(15,20)
      CALL DIG
      IXA=0
      DO 1 I=1,20
      IYA=0
      DO 2 J=1,15
      ISUM=LS(J,I)
      IF(ISUM.LT.0)ISUM=ISUM+256
      CALL ASUM(IXA,IYA,IADD)
      ISUM=ISUM+GWT*IADD/32.
      IF(ISUM.GT.127)ISUM=ISUM-256
      LS(J,I)=ISUM
2     IYA=IYA+32
1     IXA=IXA+32
      RETURN
      END
```

```
C       AUSVZ.FOR
C       PLOTTING STOPPING POINT VS. Z FOR AUZC OUTPUT.
        DIMENSION S(4),P1(6)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 9
9       FORMAT(' RANGE OF Z, RANGE OF STOPPING POINTS')
        ACCEPT 10,S(1),S(2),S(3),S(4)
10      FORMAT(7F10.0)
        IF(S(1).NE.0..OR.S(2).NE.0..OR.S(3).NE.0..OR.S(4).NE.0.)
     *GO TO 7
        IFLAG=1
        S(1)=1.E36
        DO 4 I=2,4
4       S(I)=-S(I-1)
2       READ(1,END=3)P1,IT,M,Z,SZ,SP
        GO TO (5,6),IFLAG
5       S(1)=AMIN1(S(1),Z)
        S(2)=AMAX1(S(2),Z)
        S(3)=AMIN1(S(3),SP)
        S(4)=AMAX1(S(4),SP)
        GO TO 2
6       CALL BPOINT(S,Z,SP)
        GO TO 2
3       GO TO (7,8),IFLAG
7       IFLAG=2
        CALL SCALE(S)
        REWIND 1
        GO TO 2
8       END
```

```
C       AUSZH.FOR
C       STORING SIGMA Z VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
6       FORMAT(F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        WRITE(2)SZ
        GO TO 3
4       END
```

```
C       AUTADP.FOR
C       SET UP X,Y FILE CONTAINING ALPHA AND DELTA OUTPUT BY AUTRAJ.
        DIMENSION P(6)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        TYPE 2
2       FORMAT(' OUTPUT FILE'/)
        CALL ASSIGN(2,'A',-1)
3       READ(1,END=4)P,M
        WRITE(2)57.3*P(1),57.3*P(2)
        GO TO 3
4       END
```

```
C       AUTCOM.FOR
C       COMBINING AUTRAJ OUTPUT FILES.
        DIMENSION P(6),L(100),A(100)
        TYPE 5
5       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 6
6       FORMAT(' NUMBER OF INPUT FILES')
        ACCEPT 1,N
1       FORMAT(I10)
        DO 2 I=1,N
        TYPE 7
7       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 8
8       FORMAT(' MINIMUM NUMBER OF TRACKS/PARTICLE')
        ACCEPT 1,MIN
4       READ(1,END=3)P,M,(L(J),A(J),J=1,M)
        IF(M.LT.MIN)GO TO 4
        WRITE(2)P,M,(L(J),A(J),J=1,M)
        GO TO 4
3       CALL CLOSE(1)
2       CONTINUE
        END
```

```
C       AUTDH.FOR
C       GENERATION OF A HISTOGRAM INPUT FILE FOR DELTA
C       OF TRAJECTORIES OUTPUT BY AUTRAJ.
        DIMENSION P(6)
        DPR=45./ATAN(1.)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        TYPE 2
2       FORMAT(' OUTPUT FILE'/)
        CALL ASSIGN(2,'A',-1)
3       READ(1,END=4)P,M
        WRITE(2)DPR*P(2)
        GO TO 3
4       END
```

```
C       AUTERH.FOR
C       GENERATION OF A HISTOGRAM INPUT FILE FOR THE RMS DEVIATION
C       OF TRAJECTORIES OUTPUT BY AUTRAJ.
        DIMENSION P(6)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        TYPE 2
2       FORMAT(' OUTPUT FILE'/)
        CALL ASSIGN(2,'A',-1)
3       READ(1,END=4)P,M
        WRITE(2)P(6)
        GO TO 3
4       END
```

```
C       AUTNH.FOR
C       GENERATION OF A HISTOGRAM INPUT FILE FOR THE NUMBER OF TRACKS PER
C       TRAJECTORY OUTPUT BY AUTRAJ.
        DIMENSION P(6)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        TYPE 2
2       FORMAT(' OUTPUT FILE'/)
        CALL ASSIGN(2,'A',-1)
3       READ(1,END=4)P,M
        WRITE(2)FLOAT(M)
        GO TO 3
4       END
```

```
C       AUTNHT.FOR
C       HISTOGRAM (TABLE) OF THE NUMBER OF TRAJECTORIES WITH N LAYERS.
        DIMENSION P(6),NT(100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        DO 2 I=1,100
2       NT(I)=0
3       READ(1,END=4)P,M
        NT(M)=NT(M)+1
        GO TO 3
4       DO 5 I=1,100
5       IF(NT(I).GT.0)MM=I
        DO 6 I=1,MM
6       TYPE 7,I,NT(I)
7       FORMAT(I10,I6)
        END
```

```
C       AUTPL.FOR
C       PLOTTING OUTPUT OF AUTRAJ.
        DIMENSION L(100),CH(4),Z(100),A(100),S(4)
        DATA CH/-.5,72.5,-.5,35.5/
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' MINIMUM NUMBER OF TRACKS PER PARTICLE')
        ACCEPT 6,NT
6       FORMAT(I10)
4       READ(1,END=3)AL,DE,SD,X0,Y0,SIG,N,(L(I),A(I),I=1,N)
        IF(N.LT.NT)GO TO 4
        S(1)=1.E10
        S(2)=-S(1)
        S(3)=0.
        S(4)=0.
        DO 5 I=1,N
        Z(I)=L(I)
        S(1)=AMIN1(Z(I),S(1))
        S(2)=AMAX1(Z(I),S(2))
5       S(4)=AMAX1(S(4),ABS(A(I)))
        S(4)=1.1*S(4)
        CALL SCALE(S)
        DO 9 I=1,N
        CALL LET(S,Z(I),ABS(A(I)))
        IF(A(I).GT.0.)TYPE 10,79
9       IF(A(I).LT.0.)TYPE 10,88
10      FORMAT(1X,72A1)
        CALL LET(CH,4.,35.)
        TYPE 11,57.3*AL,57.3*DE,57.3*SD,X0,Y0,SIG
D       CALL VECTOR(CH,8.,34.,72.,34.)
11      FORMAT(1X,3F10.1,2F10.2,F10.3)
        ACCEPT 7,AA
7       FORMAT(A4)
        GO TO 4
3       END
```

```
C       AUTPR.FOR
C       PRINTING THE OUTPUT OF AUTRAJ
        DIMENSION P(6),LS(100),AS(100)
        DPR=45./ATAN(1.)
        TYPE 1
1       FORMAT(' FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 8
8       FORMAT(' MINIMUM NUMBER OF LAYERS')
        ACCEPT 9,MM
9       FORMAT(7I10)
        PRINT 2
2       FORMAT('     ALPHA    DELTA      +/-        X0        Y0      SIG
     *     NUM'/'       LAYER'/' 1.E4*AREA'/)
3       READ(1,END=4)P,M,(LS(I),AS(I),I=1,M)
        IF(M.LT.MM)GO TO 3
        DO 7 I=1,3
7       P(I)=DPR*P(I)
        PRINT 5,P,M,(LS(I),I=1,M)
        PRINT 6,(INT(.5+1.E4*AS(I)),I=1,M)
5       FORMAT(/1X,3F8.1,3F8.2,I8/(1X,14I5))
6       FORMAT(1X,14I5)
        GO TO 3
4       END
```

```
C       AUTRAJ.FOR
C       FORTRAN TRACING PROGRAM.  THE RESULTS, INCLUDING THE
C       MEASURED AREA AS A FUNCTION OF LAYER ARE OUTPUT TO THE DISK.
C
C       INITIALIZING
C
        DIMENSION XA(1000),YA(1000),AA(1900),LS(100),AS(100),IV(500)
     *,DNS(10)
        DATA IR,IW,LE,PI/9,8,0,3.1415927/
        CALL SETERR(12,0)
C
C       INPUT OF PARAMETERS FROM THE CONSOLE
C
1       FORMAT(7I10)
2       FORMAT(7F10.0)
        TYPE 3
3       FORMAT(' MAX. LAYER GAP, MIN. NUMBER OF TRACKS PER TRAJECTORY')
        ACCEPT 1,LGAP,MTR
        TYPE 4
4       FORMAT (' X RANGE')
        ACCEPT 2,XMIN,XMAX
        TYPE 5
5       FORMAT (' Y RANGE')
        ACCEPT 2,YMIN,YMAX
        TYPE 43
43      FORMAT(' LAYER RANGE')
        ACCEPT 1,LA1,LA2
        TYPE 6
6       FORMAT (' MAX. ALLOWED DEVIATION
     1 FROM AN ESTABLISHED TRAJECTORY (SD)')
        ACCEPT 2,ERRM
        TYPE 12
12      FORMAT(' MINIMUM DIP ANGLE')
        ACCEPT 2,DMIN
        TYPE 33
33      FORMAT(' STANDARD RMS DEVIATION ABOUT FIT LINE')
        ACCEPT 2,SIGS
        TYPE 7
7       FORMAT (' INPUT FILE NAME'/)
        CALL ASSIGN (1,'DUM',-1,'RDO')
        TYPE 8
8       FORMAT (' OUTPUT FILE NAME'/)
        CALL ASSIGN (2,'DUM',-1,'NEW')
        TYPE 20
20      FORMAT(' STACK DATA FILE NAME'/)
        CALL ASSIGN(3,'A',-1,'RDO')
C
C       SETTING UP HEIGHT-IN-STACK FILE
C
        READ(3)NL
        DEFINE FILE 4(NL,2,U,I4)
        Z=0.
        ZL=0.
        DO 21 I=1,NL
        READ(3)H
        H=.5*H
        Z=Z+H+ZL
        WRITE(4'I)Z
21      ZL=H
C
C       COMPUTING CONSTANTS
C
        DPR=180./PI
        ERRMS=ERRM*ERRM
```

```
      DMIN=DMIN/DPR
      DMIN=COS(DMIN)/SIN(DMIN)
      SIGS=SIGS*SIGS/2.
C
C     SETTING UP SCRATCH FILES FOR TEMPORARY TRAJECTORY STORAGE
C
      CALL ASSIGN(8,'RK0:FTN8.DAT',12,'SCR')
      CALL ASSIGN(9,'RK0:FTN9.DAT',12,'SCR')
      WRITE(2)0.
      REWIND 2
      WRITE(IR)0.
      REWIND IR
      ENDFILE IR
      REWIND IR
C
C     READING THE DATA FROM ONE LAYER
C
42    READ(1)XA(1),YA(1),LN,AA(1)
      IF(XA(1).LT.XMIN.OR.XA(1).GT.XMAX.OR.YA(1).LT.YMIN.
     *OR.YA(1).GT.YMAX.OR.LN.LT.LA1)GO TO 42
      N=0
9     IF(LE.EQ.1)GO TO 39
      L=LN
      XA(1)=XA(N+1)
      YA(1)=YA(N+1)
      AA(1)=AA(N+1)
      N=2
11    READ(1,END=16)X,Y,LN,A
      IF(X.LT.XMIN.OR.X.GT.XMAX.OR.Y.LT.YMIN.
     *OR.Y.GT.YMAX)GO TO 11
      IF(LN.GT.LA2)GO TO 16
      IF(L.NE.LN)GO TO 13
      DO 14 I=N-1,1,-1
      IF(X.GT.XA(I))GO TO 15
      XA(I+1)=XA(I)
      YA(I+1)=YA(I)
14    AA(I+1)=AA(I)
      I=0
15    XA(I+1)=X
      YA(I+1)=Y
      AA(I+1)=A
      N=N+1
      GO TO 11
13    XA(N)=X
      YA(N)=Y
      AA(N)=A
      GO TO 10
16    LE=1
10    N=N-1
C
C     COMPUTING LAYER CONSTANTS
C
D     PRINT 100,(XA(I),YA(I),L,AA(I),I=1,N)
D100  FORMAT(1X,2F10.3,I4,G11.4)
      TYPE 1,L
      READ(4'L)Z
      LP=MAX0(L-LGAP-1,1)
      READ(4'LP)ZP
      DPT=DMIN*(Z-ZP)
      DPT=DPT*DPT
      ANS=.5*N
      DNS(1)=.5*ANS
      DO 26 NIT=2,10
      DNS(NIT)=.5*DNS(NIT-1)
```

```
      IF(DNS(NIT).LT.1.)GO TO 17
26    CONTINUE
      NIT=10
C
C     PROCESSING ESTABLISHED TRAJECTORIES
C
17    READ(IR,END=18)M,SZ,SZZ,SX,SZX,SXX,SY,SZY,SYY,XBIAS,YBIAS,ZBIAS,
     *(LS(I),AS(I),I=1,M)
D     PRINT 101,M,SZ,SZZ,SX,SZX,SXX,SY,SZY,SYY,XBIAS,YBIAS,ZBIAS
D101  FORMAT(1X12,5G13.6/1X,6G12.5)
      DZ=Z-ZBIAS
      IF(M-2)19,22,47
19    X=SX+XBIAS
      Y=SY+YBIAS
      DF=1.
      DP=DPT
22    SXS=SIGS
      SYS=SIGS
      IF(M.EQ.1)GO TO 48
47    AM=M
      D=AM*SZZ-SZ*SZ
      AX0=(SZZ*SX-SZ*SZX)/D
      AX1=(-SZ*SX+AM*SZX)/D
      AY0=(SZZ*SY-SZ*SZY)/D
      AY1=(-SZ*SY+AM*SZY)/D
      X=AX0+AX1*DZ+XBIAS
      Y=AY0+AY1*DZ+YBIAS
      DP=0.
      DF=(SZZ-2.*DZ*SZ+DZ*DZ*AM)/D+1.
      IF(M.EQ.2)GO TO 48
      SXS=AMAX1(SIGS,(SXX-AX0*SX-AX1*SZX)/(AM-2.))
      SYS=AMAX1(SIGS,(SYY-AY0*SY-AY1*SZY)/(AM-2.))
48    IF(L-1-LS(M).GT.LGAP)GO TO 30
      DPX=DF*SIGS*ERRMS+DP
      DPY=DF*SIGS*ERRMS+DP
      PMS=1.E37
      PNS=ANS
      DO 23 IT=1,NIT
      NS=PNS
      IF(X.GT.XA(NS))GO TO 24
      PNS=PNS-DNS(IT)
      GO TO 23
24    PNS=PNS+DNS(IT)
23    CONTINUE
      J=PNS
      DO 25 ID=-1,1,2
27    IF(J.LT.1.OR.J.GT.N)GO TO 25
      DXX=X-XA(J)
      DX=DXX*DXX
      IF(DX.LE.DPX)GO TO 49
      IF(ID*DXX)25,49,28
49    IF(AA(J).LE.0.)GO TO 28
      DY=Y-YA(J)
      DY=DY*DY
      IF(DY.GT.DPY)GO TO 28
      PS=DX/DPX+DY/DPY
      IF(PS.GE.PMS)GO TO 28
      PMS=PS
      JM=J
28    J=J+ID
      GO TO 27
25    J=PNS+1
      IF(PMS.GT.1.)GO TO 29
      X=XA(JM)-XBIAS
```

```
      Y=YA(JM)-YBIAS
      M=M+1
      LS(M)=L
      AS(M)=AA(JM)
      AA(JM)=0.
      WRITE(IW)M,SZ+DZ,SZZ+DZ*DZ,SX+X,SZX+DZ*X,SXX+X*X,
     *SY+Y,SZY+DZ*Y,SYY+Y*Y,XBIAS,YBIAS,ZBIAS,(LS(I),AS(I),I=1,M)
      GO TO 17
29    IF(L-LS(M).GT.LGAP.OR.LE.EQ.1)GO TO 30
      WRITE(IW)M,SZ,SZZ,SX,SZX,SXX,SY,SZY,SYY,XBIAS,YBIAS,ZBIAS,
     *(LS(I),AS(I),I=1,M)
      GO TO 17
30    IFLAG=1
C
C     WRITING COMPLETED TRAJECTORY TO OUTPUT FILE
C
31    IF(M.EQ.1)GO TO 32
44    ALPHA=ATAN2(AY1,AX1)
      ALPHA=AMOD(2.*PI+ALPHA,2.*PI)
      DELTA=ATAN2(1.,SQRT(AX1*AX1+AY1*AY1))
      IF(M.GT.2)GO TO 34
      SXS=SIGS
      SYS=SIGS
      GO TO 35
34    SXS=AMAX1(0.,(SXX-AX0*SX-AX1*SZX)/(AM-2.))
      SYS=AMAX1(0.,(SYY-AY0*SY-AY1*SZY)/(AM-2.))
35    SIG=SQRT(SXS+SYS)
      SAX=AM*SXS/D
      SAY=AM*SYS/D
      C=SIN(DELTA)
      IF(C.LT.1.)GO TO 45
      SIGD=SQRT(SAX+SAY)
      GO TO 36
45    SIGD=C*C*C*SQRT(SAX*AX1*AX1+SAY*AY1*AY1)/COS(DELTA)
46    GO TO 36
32    ALPHA=0.
      DELTA=PI/2.
      SIGD=DELTA
      SIG=0.
      AX0=SX
      AX1=0.
      AY0=SY
      AY1=0.
36    IF(M.GE.MTR)WRITE(2)ALPHA,DELTA,SIGD,AX0-AX1*ZBIAS+XBIAS,
     *AY0-AY1*ZBIAS+YBIAS,SIG,M,(LS(I),AS(I),I=1,M)
      GO TO (17,37),IFLAG
C
C     CREATING NEW PARTICLES FROM UNMATCHED POSITION MEASUREMENT
C     AND OTHER LAYER COMPLETION TASKS
C
18    DO 38 I=1,N
      IF(AA(I).EQ.0.)GO TO 38
C     TYPE 41,DZ,X,Y,AA(I)
C41   FORMAT(1X,5G14.7)
      WRITE(IW)1,0.,0.,0.,0.,0.,0.,0.,0.,XA(I),YA(I),Z,L,AA(I)
38    CONTINUE
      ENDFILE IW
      REWIND IR
      REWIND IW
      IR=17-IR
      IW=17-IW
      GO TO 9
C
C     WRITING REMAINING PARTICLES TO OUTPUT FILE AFTER LAST
```

```
C       MEASUREMENT INPUT
C
39      IFLAG=2
37      READ(IR,END=40)M,SZ,SZZ,SX,SZX,SXX,SY,SZY,SYY,XBIAS,YBIAS,ZBIAS,
     *(LS(I),AS(I),I=1,M)
        IF(M.EQ.1)GO TO 32
        AM=M
        D=AM*SZZ-SZ*SZ
        AX0=(SZZ*SX-SZ*SZX)/D
        AX1=(-SZ*SX+AM*SZX)/D
        AY0=(SZZ*SY-SZ*SZY)/D
        AY1=(-SZ*SY+AM*SZY)/D
        GO TO 44
40      END
```

```
C       AUXYPL.FOR
C       SETTING UP ARRAY OF X,Y POINTS FOR PLOTTING.
C       MUST BE ORDERED IN X
C       TWO FLOATING POINT VARIABLES (X,Y) PER RECORD.
        DIMENSION IO(132)
        LOGICAL*1 LO(264),IB(8)
        EQUIVALENCE (IO,LO)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        X1=1.E36
        X2=-X1
        Y1=X1
        Y2=-X1
3       READ(1,END=4)X,Y
        X1=AMIN1(X1,X)
        X2=AMAX1(X2,X)
        Y1=AMIN1(Y1,Y)
        Y2=AMAX1(Y2,Y)
        GO TO 3
4       REWIND 1
        DO 5 I=1,132
5       IO(I)=0
        IB(8)=1
        DO 6 I=7,2,-1
6       IB(I)=2*IB(I+1)
        IB(1)=-128
        SC=1200./(X2-X1)
        X0=175.-SC*X1
        Y0=1056.-SC*(Y1+Y2)*.5
        IXL=0
7       READ(1,END=8)X,Y
        IX=X0+SC*X
        IY=Y0+SC*Y
        IC=IY/8
        IBI=1+MOD(IY,8)
        IF(IX.GT.IXL)GO TO 9
10      LO(IC)=LO(IC).OR.IB(IBI)
        GO TO 7
9       WRITE(2)IO
        DO 11 I=1,132
11      IO(I)=0
        IF(IX-IXL.EQ.1)GO TO 12
        DO 13 I=IXL,IX-2
13      WRITE(2)IO
12      IXL=IX
        GO TO 10
8       END
```

```
C       AUZC.FOR
C       COMPUTATION OF Z VALUES FOR AUTOMATED HZE.
C       OUTPUT TO FILE.
C
C       INITIALIZING
C
        DIMENSION ALC(100),ZA(100),ALET(100),AN(7),WT(100)
     *,LA(100),AA(100)
        DOUBLE PRECISION H1D,H2D,HDD,B1,B2,DETD
        DATA A1,X1,AN/1.776,1.059,1.1943,-.47709,-.001314,
     *-.0014306,.00120309,-.000183602,9.4199E-06/,
     *CL,ET/3.196,.4247/
        C1=A1*CL**(1./ET)
        C2=1./(2.-X1-2./ET)
        CALL SETERR(10,255)
        CALL SETERR(12,255)
        CALL SETERR(13,255)
        CALL SETERR(14,255)
        CALL SETERR(15,255)
C
C       INPUT OF PARAMETERS FROM CONSOLE.
C
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 21
21      FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 9
9       FORMAT(' STACK DATA FILE NAME'/)
        CALL ASSIGN(3,'A',-1,'RDO')
        TYPE 34
34      FORMAT(' STANDARD ERROR IN AREA')
        ACCEPT 35,SST
35      FORMAT(7F10.0)
        SST=SST*SST
C
C       CREATING RANDOM ACCESS STACK DATA FILE
C
        READ(3)NL
        DEFINE FILE 4(NL,14,U,I4)
        Z=0.
        ZL=0.
        DO 4 I=1,NL
        READ(3)H,FW,RHO,CLET,PSI,V0,HW
        T=.5*H*FW
        Z=Z+T+ZL
        ZL=T
4       WRITE(4'I)Z,H,RHO,CLET,PSI,V0,HW
C
C       READ DATA AND CONVERT L VALUES TO WATER Z VALUES AND AREA
C       VALUES TO LET**(1./ETA) VALUES.
C
2       READ(1,END=3)ALPHA,DELTA,SDELTA,X0,Y0,SIGPOS,M
     *,(LA(I),AA(I),I=1,M)
        N=0
        DO 18 I=1,M
        N=N+1
        L=LA(I)
        READ(4'L)ZA(N),H0,RHO,CLET,PSI,V0,HW
        AA(I)=4.*AA(I)/(H0*H0)
        A=AA(I)
        IF(A.GT.0.)GO TO 12
13      N=N-1
```

```
      GO TO 18
12    CALL ETA(RHO,DELTA,0.,E,PR)
      E=E+HW*PR
      LA(N)=LA(I)
      AA(N)=AA(I)
      IF(E.GE.A)GO TO 27
      ALET(N)=0.
      GO TO 18
27    ALL=5.
      CLET=ALOG(CLET)
      FR=.2
      DO 14 J=1,100
41    ST=1./(V0+EXP(PSI*(ALL-CLET)))
      ST=AMIN1(ST,.9997)
      TT=ST/SQRT(1.-ST*ST)
      THETA=ATAN(TT)
      CALL ETA(RHO,DELTA,THETA,E,PR)
      CALL ETA(RHO,DELTA,THETA+.01,DEDT,PR1)
      E=E+HW*PR
      DEDT=DEDT+HW*PR1
      IF(DEDT.GT.0.)GO TO 40
      ALL=ALL+.5
      GO TO 41
40    DEDT=(DEDT-E)/.01
      DADL=-PSI*(1.-V0*ST)*TT*DEDT
      DL=(A-E)/DADL
      ALL=ALL+FR*DL
      FR=.5+.5*FR
C     TYPE 25,ST,A,E,DADL,ALL
      IF(ABS(DL).LT..001)GO TO 5
14    CONTINUE
      TYPE 6
6     FORMAT(' LET COMPUTATION DID NOT CONVERGE')
5     ALET(N)=EXP(-ALL/ET)
      WT(N)=DADL*DADL
18    CONTINUE
C
C     ELIMINATION OF STOPPING LAYER MEASUREMENTS
C
      IF(N.LE.3)GO TO 47
      ALM=1.E37
      DO 48 I=1,N
      IF(ALET(I).GE.ALM)GO TO 48
      ALM=ALET(I)
      NM=I
48    CONTINUE
      IF(NM.EQ.1.OR.NM.EQ.N)GO TO 47
      IF(NM.EQ.N-1.OR.ALET(N-1).EQ.0.)GO TO 49
      IF(NM.NE.2.AND.ALET(2).GT.0.)GO TO 47
      DO 50 I=2,N
      ZA(I-1)=ZA(I)
      ALET(I-1)=ALET(I)
      WT(I-1)=WT(I)
      LA(I-1)=LA(I)
50    AA(I-1)=AA(I)
49    N=N-1
C
C     COMPUTATION OF FIRST APPROXIMATION TO Z0 AND Z.
C
47    IF(N.LE.2)GO TO 2
      S1=0.
      SX=0.
      SXX=0.
      SY=0.
```

```
      SXY=0.
      SIND=SIN(DELTA)
      DO 24 I=1,N
      Y=ALET(I)
      W=WT(I)/(Y*Y)
      S1=S1+W
      SX=SX+ZA(I)*W
      SXX=SXX+ZA(I)*ZA(I)*W
      SY=SY+Y*W
      SXY=SXY+ZA(I)*Y*W
24    CONTINUE
      D=S1*SXX-SX*SX
      A=(SXX*SY-SX*SXY)/D
      B=(-SX*SY+S1*SXY)/D
      STOP=AMAX1(3.*ZA(1)-2.*ZA(N),AMIN1(3.*ZA(N)-2.*ZA(1),-A/B))
      ZN=AMAX1(6.,AMIN1(30.,(C1*SIND*ABS(B))**C2))
      IF(B)29,30,31
29    STOP=AMAX1(STOP,1.01*ZA(N)-.01*ZA(1))
      GO TO 32
30    TYPE 33
33    FORMAT(' INDETERMINATE STOPPING POINT')
      GO TO 2
31    STOP=AMIN1(STOP,1.01*ZA(1)-.01*ZA(N))
C
C     ITERATIVE CONVERGENCE TO FINAL, RIGOROUS SOLUTION.
C
32    FR=.2
      DELZL=1.E20
      DO 8 IT=1,30
      H1D=0.
      H2D=0.
      HDD=0.
      B1=0.
      B2=0.
      SIG=0.
      DO 10 I=1,N
      R=ABS(ZA(I)-STOP)/SIND
      X=ALOG(ZN*ZN*R/(A1*ZN**X1))
      AL=-2.192
      DLDR=0.
      IF(X.GE.9.093)GO TO 37
      AL=0.
      DO 11 J=7,2,-1
      AL=X*AL+AN(J)
11    DLDR=X*DLDR+(FLOAT(J)-1.)*AN(J)
      AL=X*AL+AN(1)
37    AL=AL+2.*ALOG(ZN)
      L=LA(I)
      READ(4'L)Z,H0,RHO,CLET,PSI,V0
      CLET=ALOG(CLET)
      ST=1./(V0+EXP(PSI*(AL-CLET)))
      ST=AMIN1(ST,.9997)
      TT=ST/SQRT(1.-ST*ST)
      THETA=ATAN(TT)
      CALL ETA(RHO,DELTA,THETA,E,PR)
      E=E+HW*PR
      DT=-.01
      IF(E.GT.0.)GO TO 51
      DLDR=0.
      THETA=.5*THETA
      DTH=.5*THETA
45    CALL ETA(RHO,DELTA,THETA,E,PR)
      E=E+HW*PR
      IF(DTH.LT..01)GO TO 42
```

```
      IF(E)43,43,44
43    THETA=THETA-DTH
      GO TO 46
44    THETA=THETA+DTH
46    DTH=.5*DTH
      GO TO 45
42    IF(E.GT.0.)GO TO 53
      THETA=THETA-.01
      GO TO 45
53    ST=SIN(THETA)
      TT=ST/COS(THETA)
51    IF(THETA.LT..01)DT=.01
      CALL ETA(RHO,DELTA,THETA+DT,DEDT,PR1)
      DEDT=DEDT+HW*PR1
25    FORMAT(5G14.7)
      DEDT=(DEDT-E)/DT
      E=AMAX1(0.,E)
      ALC(I)=E
      DE=(AA(I)-E)
      PSIP=PSI*(1.-V0*ST)
      DEDS=TT*PSIP*DEDT*DLDR/(Z-STOP)
      DEDZ=PSIP*TT*((XI-2.)*DLDR-2.)*DEDT/ZN
D     TYPE 25,R,E,DE,DEDZ,DEDS,TT,PSIP,DEDT,DLDR
      H1D=H1D+DEDS*DEDS
      H2D=H2D+DEDZ*DEDZ
      HDD=HDD+DEDS*DEDZ
      B1=B1+DE*DEDS
      B2=B2+DE*DEDZ
      SIG=SIG+DE*DE
10    CONTINUE
D     TYPE 36,H1D,H2D,HDD,B1,B2
D36   FORMAT(1X,5D14.7)
      IF(H1D.EQ.0.)GO TO 38
      DETD=H1D*H2D-HDD*HDD
      DELS=(H2D*B1-HDD*B2)/DETD
      DELZ=(-HDD*B1+H1D*B2)/DETD
      GO TO 39
38    DELS=0.
      DELZ=B2/H2D
      DETD=H2D
      H1D=1.
      H2D=0.
39    H1=H1D
      H2=H2D
      HD=HDD
      DET=DETD
      STOPP=STOP
      ZNP=ZN
      IF(DELZ/DELZL.LT.-.5)FR=.2*FR
      DELZL=DELZ
      STOP=STOP+FR*DELS
      ZN=ZN+FR*DELZ
D     TYPE 25,ZN,STOP,DELZ,DELS,FR
      FR=.5+.5*FR
      IF(ZN.GE.0.)GO TO 26
      ZN=.5*ZNP
      FR=.2
26    IF(B.GT.0..AND.STOP.GT.ZA(1))
     *STOP=AMIN1(ZA(1)-.01,.5*(ZA(1)+STOPP))
      IF(B.LT.0..AND.STOP.LT.ZA(N))
     *STOP=AMAX1(ZA(N)+.01,.5*(ZA(N)+STOPP))
      IF(ABS(DELS).LT..01.AND.ABS(DELZ).LT..01)GO TO 16
8     CONTINUE
C
```

```
C       COMPUTATION OF ERRORS AND OUTPUT OF THE RESULTS TO THE
C       OUTPUT FILE.
C
16      D=N-2
        D=AMAX1(D,1.)
        SIG=SIG/D
        SG=AMAX1(SIG,0.)
        SIG=AMAX1(SIG,SST)
        HP=AMAX1(H1*H2,.3E-38)
        WRITE(2)ALPHA,DELTA,SDELTA,X0,Y0,SIGPOS,IT,N
     *,ZN,SQRT(ABS(SIG*H1/DET)),STOP,SQRT(ABS(SIG*H2/DET)),
     *-HD/SQRT(HP),SQRT(SG),(ZA(I),AA(I),
     *ALC(I),I=1,N)
        TYPE 15,57.3*ALPHA,57.3*DELTA,57.3*SDELTA,X0,Y0,SIGPOS,IT,N
     *,ZN,SQRT(ABS(SIG*H1/DET)),STOP,SQRT(ABS(SIG*H2/DET)),
     *-HD/SQRT(HP),SQRT(SG)
15      FORMAT(/1X,'ALPHA=',F6.1,', DELTA=',F4.1,'+/-',F4.1,', X0=',F6.2,
     *', Y0=',F6.2/1X,'SIG.POS.=',F4.2,', ITT.=',I2,', NUM.PTS.=',I2
     */1X,'Z=',F4.1,'+/-',F3.1,', ST.PT.=',
     *F5.1,'+/-',F5.1,' CORR.=',F6.3,' RMS.DEV.=',F6.4)
        GO TO 2
3       TYPE 52,(7,I=1,10)
52      FORMAT(1X,72A1)
        END
```

```
C       AUZCG.FOR
C       PLOTTING OUTPUT OF AUZC.
        DIMENSION SL(4),S(4),Z(100),AM(100),AC(100)
        DATA SL/.5,72.5,.5,35.5/
        CALL SETERR(10,0)
        CALL SETERR(11,0)
        CALL SETERR(12,0)
        CALL SETERR(13,0)
        CALL SETERR(14,0)
        CALL SETERR(15,0)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
D       TYPE 2
D2      FORMAT(' OUTPUT FILE NAME'/)
D       CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 9
9       FORMAT(' RANGE OF ACCEPTABLE NUMBER OF TRACKS/PARTICLE')
        ACCEPT 10,N1,N2
10      FORMAT(2I10)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,ZN,SZ,ST,SST,C,RMS
     *,(Z(I),AM(I),AC(I),I=1,N)
        IF(N.LT.N1.OR.N.GT.N2)GO TO 3
        S(1)=1.E10
        S(2)=-S(1)
        S(3)=0.
        S(4)=0.
        DO 5 I=1,N
        S(1)=AMIN1(S(1),Z(I))
        S(2)=AMAX1(S(2),Z(I))
5       S(4)=AMAX1(S(4),AM(I),AC(I))
        S(4)=1.15*S(4)
        CALL SCALE(S)
        CALL LET(S,Z(1),AM(1))
        TYPE 8
8       FORMAT(' X')
        DO 6 I=2,N
        CALL LET(S,Z(I),AM(I))
        TYPE 8
6       CALL VECTOR(S,Z(I-1),AC(I-1),Z(I),AC(I))
        CALL LET(SL,1.,35.)
        TYPE 15,57.3*A,57.3*D,57.3*SD,X,Y,SP,IT,N,ZN,SZ,ST,SST,C,RMS
        ACCEPT 7,I
7       FORMAT(A2)
        GO TO 3
15      FORMAT(11X,'ALPHA=',F6.1,', DELTA=',F4.1,'+/-',F4.1,', X0=',
     *F6.2,', Y0=',F6.2/11X,'SIG.POS.=',F4.2,', ITT.=',I2,', NUM.PTS.=',
     *I2/11X,'Z=',F4.1,'+/-',F3.1,', ST.PT.=',
     *F5.1,'+/-',F5.1,' CORR.=',F6.3,' RMS.DEV.=',F6.4)
4       END
```

```
C       AUZCOM.FOR
C       COMBINING AUZC OUTPUT FILES.
        DIMENSION P1(6),P2(6),Z(100),A(100),AC(100)
        TYPE 5
5       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1)
        TYPE 6
6       FORMAT(' NUMBER OF INPUT FILES')
        ACCEPT 1,N
1       FORMAT(I10)
        DO 2 I=1,N
        TYPE 7
7       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 8
8       FORMAT(' MINIMUM NUMBER OF TRACKS/PARTICLE')
        ACCEPT 1,MIN
4       READ(1,END=3)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        IF(M.LT.MIN.OR.Z(M)-Z(1).GT.20.)GO TO 4
        WRITE(2)P1,IT,M,P2,(Z(J),A(J),AC(J),J=1,M)
        GO TO 4
3       CALL CLOSE(1)
2       CONTINUE
        END
```

```
C       AUZCP.FOR
C       PRINTING OUTPUT OF AUZC.
        DIMENSION Z(50),AM(50),AC(50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,ZN,SZ,ST,SST,C,RMS
     *,(Z(I),AM(I),AC(I),I=1,N)
        TYPE 15,57.3*A,57.3*D,57.3*SD,X,Y,SP,IT,N,ZN,SZ,ST,SST,C,RMS
     *,(Z(I),AM(I),AC(I),AM(I)-AC(I),I=1,N)
        GO TO 3
15      FORMAT(/1X,'ALPHA=',F6.1,', DELTA=',F4.1,'+/-',F4.1,', X0=',F6.2,
     *', Y0=',F6.2/1X,'SIG.POS.=',F4.2,', ITT.=',I2,', NUM.PTS.=',I2
     */1X,'Z=',F4.1,'+/-',F3.1,', ST.PT.=',
     *F5.2,'+/-',F5.2,' CORR.=',F6.3,' RMS.DEV.=',F6.4/'       Z
     *        A       AC     A-AC'/(1X,F6.2,3F8.4))
4       END
```

```
C       AUZCPS.FOR
C       PRINTING OUTPUT OF AUZC.
C       SHORT FORM WITH NO INDIVIDUAL TRACK PARAMETERS
        DIMENSION Z(50),AM(50),AC(50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,ZN,SZ,ST,SST,C,RMS
     *,(Z(I),AM(I),AC(I),I=1,N)
        TYPE 15,57.3*A,57.3*D,57.3*SD,X,Y,SP,IT,N,ZN,SZ,ST,SST,C,RMS
        GO TO 3
15      FORMAT(/1X,'ALPHA=',F6.1,', DELTA=',F4.1,'+/-',F4.1,', X0=',F6.2,
     *', Y0=',F6.2/1X,'SIG.POS.=',F4.2,', ITT.=',I2,', NUM.PTS.=',I2
     */1X,'Z=',F4.1,'+/-',F3.1,', ST.PT.=',
     *F5.1,'+/-',F5.1,' CORR.=',F6.3,' RMS.DEV.=',F6.4)
4       END
```

```
C       AUZER3.FOR
C       Z ERROR CALCULATION FOR A SEMI-INFINITE STACK COMPOSED OF
C       REPEAT UNITS MADE UP OF CN, LEXAN, AND CR-39 LAYERS.
        DIMENSION S(4),SZ(26),REG(3),B1(3),CP(3),BC(3),PPAS(3)
        COMMON NL(3),FR(3),H(3),F(3),RR,V0(3),CLET(3),PSI(3)
        DATA V0/0.,0.,1./,CLET/35.56,259.1,157./,PSI/2.425,2.104,1.75/,
     *F/1.302,1.117,1.27/,PI/3.1415927/
        DATA B1/4.6E-5,5.4E-5,1.9E-5/,CP/-.17,-.277,-.17/
        CALL SETERR(12,0)
        TYPE 9
9       FORMAT(' NUMBER OF INTEGRATION INTERVALS')
        ACCEPT 2,NI
        TYPE 10
10      FORMAT(' Z INTERVAL')
        ACCEPT 2,IZI
        TYPE 20
20      FORMAT(' SCALE FACTOR (MM/PP)')
        ACCEPT 11,SF
8       TYPE 1
1       FORMAT(' IN CN: NUMBER OF LAYERS/UNIT, LAYER THICKNESS (MM),'/
     *,' FRACTION OF LAYER REMOVED DURING ETCH')
        ACCEPT 2,NL(1),H(1),FR(1)
2       FORMAT(I10,2F10.0)
        TYPE 3
3       FORMAT(' IN LEXAN: NUMBER OF LAYERS/UNIT, LAYER THICKNESS (MM),'/
     *,' FRACTION OF LAYER REMOVED DURING ETCH')
        ACCEPT 2,NL(2),H(2),FR(2)
        TYPE 18
18      FORMAT(' IN CR-39: NUMBER OF LAYERS/UNIT, LAYER THICKNESS(MM),'/
     *,' FRACTION OF LAYER REMOVED DURING ETCH')
        ACCEPT 2,NL(3),H(3),FR(3)
        RR=0.
        DO 4 I=1,3
        IF(NL(I).EQ.0)GO TO 4
        PPAS(I)=2.*SF/H(I)
        PPAS(I)=PPAS(I)*PPAS(I)*PPAS(I)/12.
        BC(I)=SQRT(2.*B1(I)/H(I))
4       RR=RR+NL(I)*H(I)*F(I)
        DO 5 I=1,3
5       H(I)=F(I)*H(I)/RR
        TYPE 7
7       FORMAT(' DIP ANGLE')
        ACCEPT 11,DELTA
11      FORMAT(7F10.0)
        TYPE 17
17      FORMAT(' MAX. DZ (MM OF WATER)')
        ACCEPT 11,ZMAX
        DPR=180./PI
        DELTA=DELTA/DPR
        SD=SIN(DELTA)
        SSD=SQRT(SD)
        DO 19 I=1,3
19      BC(I)=BC(I)*SSD
        S(1)=6.
        S(2)=26.
        S(3)=0.
        S(4)=-1.E37
        DO 6 IZ=6,26,IZI
        Z=IZ
        CALL RREG(Z,DELTA,REG)
        SPS=0.
        SPZ=0.
        SPD=0.
        ALETC=4.079*Z**1.6
```

```
      DO 12 I=1,3
      IF(NL(I).EQ.0)GO TO 12
      CALL ETA(FR(I),DELTA,0.,A,P)
      IF(A.EQ.0.)GO TO 12
      R=H(I)/SD
      IF(R.GT.REG(I))GO TO 12
      DR=(AMIN1(ZMAX/(RR*SD),REG(I))-R)/NI
      W=DR*NL(I)
      DO 13 II=0,NI
      ALET=ALETC/(R*RR)**.4247
      ST=1./(V0(I)+(ALET/CLET(I))**PSI(I))
D     TYPE 15,R,ALET,ST
D15   FORMAT(5G14.7)
      CT=SQRT(1.-ST*ST)
      TT=ST/CT
      T=ATAN2(ST,CT)
      TP=T+.01
      IF(II.EQ.NI)TP=T-.01
      CALL ETA(FR(I),DELTA,T,A,P)
      CALL ETA(FR(I),DELTA,TP,AP,P)
      DADT=(AP-A)/(TP-T)
      DADS=-PSI(I)*TT*ST*(ALET/CLET(I))**PSI(I)*DADT
      DADZ=1.6*DADS/Z
      DADS=-.4247*DADS/R
      SA=ST**CP(I)*TT*BC(I)*DADT
      SA=1./(SA*SA+PPAS(I)*P)
      WP=2.
      IF(MOD(II,2).EQ.1)WP=4.
      IF(II.EQ.0.OR.II.EQ.NI)WP=1.
D     TYPE 15,A,DADZ,DADS,SA,DADT
      WP=WP*W*SA
      SPS=SPS+WP*DADS*DADS
      SPZ=SPZ+WP*DADZ*DADZ
      SPD=SPD+WP*DADS*DADZ
13    R=R+DR
12    CONTINUE
      IF(SPS.NE.0.)GO TO 16
      SZ(IZ)=0.
      GO TO 6
16    SPS=SD*SPS/3.
      SPZ=SPZ*SD/3.
      SPD=SPD*SD/3.
D     TYPE 15,SPS,SPZ,SPD
      SZ(IZ)=SQRT(SPS/(SPZ*SPS-SPD*SPD))
      S(4)=AMAX1(S(4),SZ(IZ))
6     CONTINUE
      YL=0.
      S(4)=AMIN1(S(4),9.9)
      CALL SCALE(S)
      DO 14 I=6,26,IZI
      X=I
      Y=SZ(I)
      IF(YL.NE.0.)CALL VECTOR(S,X,Y,XL,YL)
      XL=X
14    YL=Y
      ACCEPT 2,I
      GO TO 8
      END
C
C
      SUBROUTINE RREG(Z,DELTA,R)
C
      DIMENSION R(3)
      COMMON NL(3),FR(3),H(3),F(3),RR,V0(3),CLET(3),PSI(3)
```

```
      PB2=1.570796
      RZ=27.39*Z**3.767/RR
      DO 1 I=1,3
      R(I)=0.
      SD=SIN(DELTA)
      IF(DELTA.GE.PB2)GO TO 2
      CD=COS(DELTA)
      TH=2.*ATAN(FR(I)*SD/CD)-DELTA
      IF(TH.LE.0.)GO TO 1
      IF(DELTA+TH-PB2)4,4,2
4     ST=SIN(TH)
      GO TO 3
2     ST=FR(I)*SD
3     ALET=CLET(I)*(1./ST-V0(I))**(1./PSI(I))
      R(I)=RZ/ALET**2.355
1     CONTINUE
      RETURN
      END
```

```
C       AUZH.FOR
C       STORING Z VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,100)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 5
5       FORMAT(' ERROR LIMITS')
        ACCEPT 6,EL1,EL2
6       FORMAT(7F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        IF(C.NE.0..AND.SZ.LE.EL2.AND.SZ.GE.EL1)WRITE(2)Z
        GO TO 3
4       END
```

```
C       AUZNH.FOR
C       STORING AUZC OUTPUT N VALUES FOR HISTOGRAM.
        DIMENSION PAR(3,50)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'A',-1,'NEW')
        TYPE 5
5       FORMAT(' ERROR LIMIT')
        ACCEPT 6,EL
6       FORMAT(F10.0)
3       READ(1,END=4)A,D,SD,X,Y,SP,IT,N,Z,SZ,ST,SST,C,RMS,
     *((PAR(I,J),I=1,3),J=1,N)
        IF(SZ.LT.EL)WRITE(2)FLOAT(N)
        GO TO 3
4       END
```

```
      SUBROUTINE BPOINT (S,A,B)
      DIMENSION S(4)
      IF (A.LT.S(1).OR.A.GT.S(2).OR.B.LT.S(3).OR.B.GT.S(4))
    1 RETURN
      X=1023.*(A-S(1))/(S(2)-S(1))
      Y=779.*(B-S(3))/(S(4)-S(3))
      IX=INT(X/32.)
      IY=INT(Y/32.)
      TYPE 1,28,32+IY,96+INT(Y)-32*IY,32+IX,64+INT(X)-32*IX,30,81,86,90,89
    1 FORMAT(1H+,10A1)
      RETURN
      END
```

```
        .TITLE COLUMN
;FORTRAN CALLABLE SUBROUTINE COLUMN(LC,IC,N,LT,LH)
;WHERE LC(15) IS THE GRAY LEVEL CUTOFF ARRAY (ONE ELEMENT
;PER 32 IN THE Y ADDRESS), IC IS THE X ADDRESS OF THE COLUMN,
;N IS THE NUMBER OF DETECTED LINE SEGMENTS, LT IS THE ARRAY WITH
;THE ADDRESSES OF THE TOP OF THE SEGMENTS, AND LH IS THE ARRAY CON-
;TAINING THE HALF HEIGHTS OF THE SEGMENTS.
;
        .MCALL .REGDEF
        .REGDEF
        .GLOBL COLUMN
;
COLUMN: CLR @6(R5)              ;SEGMENT COUNT
        MOV 2(R5),R4            ;GET STARTING CUTOFF ADDRESS
        DEC R4
        MOV #177776,R0          ;Y ADDRESS
        MOV @4(R5),@#164204     ;LOAD X REGISTER
        MOV 10(R5),R2           ;LT POINTER
        MOV 12(R5),R3           ;LH POINTER
;
;UNDETECTED REGION
U1:     MOV #15.,R1             ;LOAD CUTOFF COUNTER
        INC R4                  ;INCREMENT CUTOFF ADDRESS
        CMP R0,#478.            ;IS THE COLUMN COMPLETE
        BMI U2                  ;NO
        RTS PC                  ;YES. RETURN
U2:     TST (R0)+               ;INCREMENT Y ADDRESS
        MOV R0,@#164206         ;LOAD Y REGISTER
        CMPB @#164236,(R4)              ;IS POINT DETECTED
        BHI D4                  ;YES. BRANCH TO DETECTED SECTION
U3:     DEC R1                  ;DECREMENT CUTOFFF COUNTER
        BPL U2                  ;IF CUTOFF AREA NOT COMPLETE RETAIN
                                ;CUTOFF VALUE
        BR U1                   ;CUTOFF AREA COMPLETE. GET NEW CUTOFF
                                ;VALUE
;
;CHANGE TO UNDETECTED REGION
U4:     TST (R3)+               ;UPDATE LH POINTER
        TST -(R0)               ;DECREMENT Y ADDRESS
        MOV R0,@#164206         ;LOAD Y REGISTER
        CLR @#164236            ;MAKE POINT BLACK
        TST (R0)+               ;INCREMENT Y ADDRESS
        CMP @6(R5),#20.         ;BUFFER FULL?
        BLT U3                  ;NO, CONTINUE UNDETECTED REGION
        INC @6(R5)              ;YES, INCREMENT COUNT TO INDICATE OVERFLOW
        RTS PC                  ;ABORTIVE RETURN
;
;DETECTED REGION (SEE COMENTS FROM UNDETECTED REGION)
D1:     MOV #15.,R1
        INC R4
        CMP R0,#478.
        BMI D2
;
;DETECTED REGION RETURN
        MOV R0,@#164206
        CLR @#164236
        RTS PC
D2:     TST (R0)+
        MOV R0,@#164206
        CMPB @#164236,(R4)
        BLOS U4
        INC (R3)
D3:     DEC R1
        BPL D2
```

```
        BR D1
;
;CHANGE TO DETECTED REGION
D4:     MOV #1,(R3)
        INC @6(R5)
        MOV R0,(R2)+
        CLR @#164236
        BR D3
        .END COLUMN
```

```
        .TITLE DIG
;FORTRAN CALLABLE SUBROUTINE FOR THE
;DIGITIZATION OF EYECOM FIELD WITH HIGH SPEED DIGITIZER
        .GLOBL DIG
DIG:    MOV #40,@#164200
A:      BIT #40,@#164200
        BNE A
        MOV #1000,@#164200
        MOV #1000,@#164200
        MOV #141,@#164202
        RTS %7
        .END DIG
```

```
C       ETA.FOR
        SUBROUTINE ETA(R2,DELTA,THETA,AREA,PERIM)
C       SUBROUTINE TO COMPUTE PROJECTED ETCH THROUGH AREA AND PERIMETER.
C       2B/H0=R2
        DIMENSION AA(2),BA(2),CA(2),XA(3)
        R7=COS(THETA)
        R8=SIN(THETA)
        S5=COS(DELTA)
        S6=SIN(DELTA)
        S3=0.
        AREA=0.
        PERIM=0.
        S1=(R2-R8/S6)/R7
        IF( S1.GT.0.)GO TO 20
        GO TO 1
20      R9=S1*S6
        Z=R8+S6
        S0=R2*R7/Z
        S7=R9
        XA(1)=R9
        S4=S5/S6-R2*S5/Z
        C=S6-R8
        S2=R2*SQRT(C/Z)
        IF( S1*S5.GT.1.-R2 )GO TO 50
        K=1
        IF( 1.570796327.LE.DELTA+THETA )GO TO 80
        K=2
        S7=S0-S4
        XA(1)=S7
        GO TO 80
50      R9=R2*R7/C-2.*R7*R8/(Z*C)
        S1=R2*SQRT(Z/C)-2.*R8/SQRT(Z*C)
        S3=S5*((S6*S6+R8*R8)/Z-R2*S6)/(S6*C)
        K=1
        S7=R9-S3
        J=2
        IF(S4.NE.S3)S8=(S0*S0-R9*R9-S4*S4+S3*S3)/(2.*(S4-S3))
        XA(1)=S7
        XA(2)=S8
        IF( S8.GT.0.)GO TO 130
        J=1
        GO TO 130
80      IF( S7.GT.0.)GO TO 90
        GO TO 1
90      C=S2/S0
        Z=S1/R9
        A=Z*Z-C*C
        B=-2.*S4*C*C
        C=C*S4
        C=S2*S2-S1*S1-C*C
        J=1
        Z=B*B-4*A*C
        IF( Z.LE.0.)GO TO 130
        Z=SQRT(Z)
        X=(-B+Z)/(2.*A)
        IF( S7.LE.X )GO TO 120
        IF( X.LE.0.)GO TO 120
        J=2
        S8=X
        XA(2)=X
120     X=(-B-Z)/(2.*A)
        IF( XA(J).LE.X )GO TO 130
        IF( X.LE.0.)GO TO 130
        J=J+1
```

```
      XA(J)=X
130   XA(1+J)=0.
      AA(1)=R9
      AA(2)=S0
      BA(1)=S1
      BA(2)=S2
      CA(1)=S3
      CA(2)=S4
      DO 202 I=1 , J
      A=AA(K)
      B=BA(K)
      C=CA(K)
      K=3-K
      X=(XA(1+I)+C)/A
      Y=(XA(I)+C)/A
      IF(Y.GT.1.)Y=1.
      IF(X.GT.1.)X=1.
154   AREA=AREA+2.*A*B*(Y*SQRT(1.-Y*Y)-X*SQRT(1.-X*X))
      X=ATAN2(X,SQRT(1.-X*X))
      IF( Y.EQ.1. )GO TO 163
      Y=ATAN(Y/SQRT(1.-Y*Y))
      GO TO 164
163   Y=1.570796327
164   AREA=AREA+2.*A*B*(Y-X)
      C=1.-B*B/(A*A)
      Y=(Y-X)/2
      X=X+Y
      A=4.*A*Y
      B=SIN(X)
      PERIM=PERIM+A*SQRT(1.-C*B*B)/1.125
      S5=.7745966692
      B=SIN(X+S5*Y)
190   B=SQRT(1.-C*B*B)
      X=SIN(X-S5*Y)
      B=B+SQRT(1.-C*X*X)
      PERIM=PERIM+A*B/1.8
202   CONTINUE
1     RETURN
      END
```

```
C       ETAPLT.FOR
C       PLOTTING THE ETCHED THROUGH AREA AS A FUNCION OF THETA
C       AND THE PARAMETERS 2B/H0 AND DELTA.
        DIMENSION S(4),AREA(101)
        DPR=45./ATAN(1.)
1       TYPE 2
2       FORMAT(' 2B/H0, DELTA(DEGREES),THETA RANGE(DEGREES)')
        ACCEPT 3,R,DELTA,THETA1,THETA2
3       FORMAT(4F10.5)
        DELTA=DELTA/DPR
        S(1)=THETA1
        S(2)=THETA2
        S(3)=999.
        S(4)=0.
        DT=(THETA2-THETA1)/100.
        DO 4 I=1,101
        CALL ETA(R,DELTA,(THETA1+(I-1)*DT)/DPR,AREA(I),PERIMD
        IF(AREA(I).LT.S(3))S(3)=AREA(I)
        IF(AREA(I).GT.S(4))S(4)=AREA(I)
4       CONTINUE
        CALL SCALE(S)
        DO 5 I=1,101
        CALL VECTOR(S,THETA1+(I-1)*DT,AREA(I),THETA1+I*DT,AREA(I+1))
5       CONTINUE
        ACCEPT 6,I
6       FORMAT(A2)
        GO TO 1
        END
```

```
C       ETAPRT.FOR
C       PRINTING THE AREA AND PERIMETER A COMPUTED BY ETA.FOR.
C       RHO=2.*B/H0
1       TYPE 2
2       FORMAT(' RHO')
        ACCEPT 3,RHO
3       FORMAT(7F10.0)
4       TYPE 5
5       FORMAT(' DELTA')
        ACCEPT 3,DELTA
        IF(DELTA.EQ.0.)GO TO 1
6       TYPE 7
7       FORMAT(' THETA')
        ACCEPT 3,THETA
        IF(THETA.EQ.0.)GO TO 4
        CALL ETA(RHO,DELTA,THETA,AREA,PERIM)
        TYPE 8,AREA,PERIM
8       FORMAT(1X,'AREA =',F7.4,5X,'PERIMETER =',F7.4)
        GO TO 6
        END
```

```
C       ETAVRP.FOR
C       PLOTTING THE ETCHED THROUGH AREA AS A FUNCION OF RES. RANGE
C       AND THE PARAMETERS 2B/H0 AND DELTA.
        DIMENSION S(4),AREA(100)
        DPR=45./ATAN(1.)
1       TYPE 2
2       FORMAT(' 2B/H0, DELTA(DEGREES),
     * LAMBDA')
        ACCEPT 3,R,DELTA,ALAM
3       FORMAT(5F10.5)
        ALAM=1./ALAM
        DELTA=DELTA/DPR
        S(1)=0.
        S(3)=999.
        S(4)=0.
        DT=.01
        DO 4 I=1,100
        THETA=(I*DT)**ALAM
        IF(THETA.GE.1.)THETA=.9999999
        THETA=ATAN2(THETA,SQRT(1.-THETA*THETA))
        CALL ETA(R,DELTA,THETA,AREA(I),PERIM)
        IF(AREA(I).LT.S(3))S(3)=AREA(I)
        IF(AREA(I).GT.S(4))S(4)=AREA(I)
        N=I
        IF(AREA(I).LE.0.)GO TO 7
4       CONTINUE
7       S(2)=.01*N
        CALL SCALE(S)
        DO 5 I=1,N-1
        CALL VECTOR(S,I*DT,AREA(I),.01+I*DT,AREA(I+1))
5       CONTINUE
        ACCEPT 6,I
6       FORMAT(A2)
        GO TO 1
        END
```

```
C       ETPPLT.FOR
C       PLOTTING THE ETCHED THROUGH PERIMETER AS A FUNCION OF THETA
C       AND THE PARAMETERS 2B/H0 AND DELTA.
        DIMENSION S(4),AREA(101)
        DPR=45./ATAN(1.)
1       TYPE 2
2       FORMAT(' 2B/H0, DELTA(DEGREES),THETA RANGE(DEGREES)')
        ACCEPT 3,R,DELTA,THETA1,THETA2
3       FORMAT(4F10.5)
        DELTA=DELTA/DPR
        S(1)=THETA1
        S(2)=THETA2
        S(3)=999.
        S(4)=0.
        DT=(THETA2-THETA1)/100.
        DO 4 I=1,101
        CALL ETA(R,DELTA,(THETA1+(I-1)*DT)/DPR,PERIM,AREA(I))
        IF(AREA(I).LT.S(3))S(3)=AREA(I)
        IF(AREA(I).GT.S(4))S(4)=AREA(I)
4       CONTINUE
        CALL SCALE(S)
        DO 5 I=1,101
        CALL VECTOR(S,THETA1+(I-1)*DT,AREA(I),THETA1+I*DT,AREA(I+1))
5       CONTINUE
        ACCEPT 6,I
6       FORMAT(A2)
        GO TO 1
        END
```

```
      SUBROUTINE FEATUR(LS,N,XP,YP,AREA)
C     DETECTION AND MEASURMENT OF FEATURES WITH EYECOM.
C     N=NUMBER OF FEATURES FOUND.
C     XP=ARRAY OF X COORDINATES.
C     YP=ARRAY OF Y COORDINDATES.
C     AREA=ARRAY OF TRACK AREAS
C     LS=ARRAY OF GRAY CUTOFF VALUES
      DIMENSION XP(200),YP(200),AREA(200),LT(200),LH(200),NT(20),
     *NH(20),NF(20)
      LOGICAL*1 LS(15,20)
      N=0
      NM=1
      DO 1 I=1,639,2
      IGC=I/32+1
      CALL COLUMN(LS(1,IGC),I,M,NT,NH)
D     TYPE 25,IPEEK("164206)
D25   FORMAT(1X,7I10)
      IF(M.LE.20)GO TO 19
      M=20
      TYPE 20,I
20    FORMAT(' TOO MANY SEGMENTS IN COLUMN',I4)
19    IF(M.GT.0)GO TO 2
      IF(NM.GT.N)GO TO 1
      DO 14 J=NM,N
14    AREA(J)=ABS(AREA(J))
      NM=N+1
      GO TO 1
2     IF(N.GE.NM)GO TO 3
      M=MIN0(M,200-N)
      DO 4 J=1,M
      NP=N+J
      ANH=NH(J)
      XP(NP)=I*ANH
      YP(NP)=ANH*(NT(J)+ANH)
      AREA(NP)=-ANH
      LT(NP)=NT(J)
4     LH(NP)=NH(J)
      N=N+M
      IF(N.LT.200)GO TO 1
23    TYPE 21
21    FORMAT(' COUNT FULL')
      GO TO 22
3     DO 6 J=1,M
6     NF(J)=0
      DO 5 J=NM,N
      IF(AREA(J).GE.0.)GO TO 5
      ISF=0
      DO 7 K=1,M
      IF(NT(K)+2*NH(K).LT.LT(J))GO TO 7
      IF(NT(K).GT.LT(J)+2*LH(J))GO TO 8
      ANH=NH(K)
      XP(J)=XP(J)+I*ANH
      YP(J)=YP(J)+ANH*(NT(K)+ANH)
      AREA(J)=AREA(J)-ANH
      IF(ISF.GT.0)GO TO 9
      ISF=1
      IF(NF(K).GT.0)GO TO 10
11    LT(J)=NT(K)
      LH(J)=NH(K)
      NF(K)=J
      GO TO 7
10    L=NF(K)
      XP(J)=XP(J)+XP(L)
      YP(J)=YP(J)+YP(L)
```

```
      AREA(J)=AREA(J)+AREA(L)
      AREA(L)=0.
      GO TO 11
9     LH(J)=(NT(K)+2*NH(K)-LT(J))/2
      NF(K)=J
7     CONTINUE
8     IF(ISF.EQ.0)AREA(J)=ABS(AREA(J))
5     CONTINUE
      DO 12 J=NM,N
12    IF(AREA(J).LT.0.)GO TO 13
13    NM=J
      DO 15 K=1,M
      IF(NF(K).GT.0)GO TO 15
      N=N+1
      ANH=NH(K)
      XP(N)=I*ANH
      YP(N)=ANH*(NT(K)+ANH)
      AREA(N)=-ANH
      LT(N)=NT(K)
      LH(N)=NH(K)
      IF(N.EQ.200)GO TO 23
15    CONTINUE
1     CONTINUE
22    NP=0
      IF(N.EQ.0)GO TO 24
      DO 16 I=1,N
      IF(AREA(I))17,16,18
17    AREA(I)=-AREA(I)
18    NP=NP+1
      XP(NP)=XP(I)/AREA(I)
      YP(NP)=YP(I)/AREA(I)
      AREA(NP)=AREA(I)
16    CONTINUE
24    N=NP
      RETURN
      END
```

```
C       FX.FOR
        SUBROUTINE FX(N,X,F)
        DIMENSION F(20)
        F(1)=1.
        IF(N.EQ.1)RETURN
        DO 1 I=2,N
1       F(I)=F(I-1)*X
        RETURN
        END
```

```
C       CHIST.FOR
C       PLOTTING HISTOGRAM CONSISTING OF A SUM OF GAUSSIANS.
        DIMENSION S(4),X(600),Y(600)
1       TYPE 2
2       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
3       REWIND 1
15      TYPE 4
4       FORMAT(' X RANGE, INTERVAL, SIGMA')
        ACCEPT 5,X1,X2,XI,SIG
5       FORMAT(7F10.0)
        N=(X2-X1)/XI+1.5
        IF(N.LE.600)GO TO 13
        TYPE 14
14      FORMAT(' TOO MANY POINTS.')
        GO TO 15
13      C=.3989423/SIG
        TYPE 16
16      FORMAT(' DATA SCALE FACTOR')
        ACCEPT 5,SC
        DO 7 I=1,N
        Y(I)=0.
7       X(I)=X1+XI*FLOAT(I-1)
8       READ(1,END=9)XV
        XV=XV*SC
        DO 6 I=1,N
        D=ABS(XV-X(I))/SIG
        IF(D.GT.4.)GO TO 6
        Y(I)=Y(I)+C/EXP(D*D*.5)
6       CONTINUE
        GO TO 8
9       S(1)=X(1)
        S(2)=X(N)
        S(3)=0.
        S(4)=0.
        DO 10 I=1,N
10      S(4)=AMAX1(S(4),Y(I))
        CALL SCALE(S)
        DO 11 I=2,N
11      CALL VECTOR(S,X(I-1),Y(I-1),X(I),Y(I))
        ACCEPT 12,I
12      FORMAT(A2)
        GO TO 3
        END
```

```
C       GRAPLT.FOR
C       GRAY LEVEL PLOT FROM UNFORMATTED SEQUENTIAL FILE.
        DIMENSION IP(132)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        CALL MTX(IP,0,2)
2       READ(1,END=3)IP
        CALL MTX(IP,132,2)
        GO TO 2
3       DO 4 I=1,132
4       IP(I)=0
        DO 5 I=1,300
        DO 6 J=1,800
6       CONTINUE
5       CALL MTX(IP,132,2)
        CALL MTX(IP,0,2)
        END
```

```
      SUBROUTINE HAIRS(S,X,Y,ICHAR)
C     DISPLAYING AND READING OUT POSITION OF CROSS HAIRS FROM A SCALED
C     FIELD.
      DIMENSION S(4)
      LOGICAL*1 D(6)
1     FORMAT('$',72A1)
      TYPE 1,27,26
      ACCEPT 2,(D(I),I=1,5)
2     FORMAT(5A1)
      X=(32*D(2)+D(3)-1056)*(S(2)-S(1))/1023.+S(1)
      Y=(32*D(4)+D(5)-1056)*(S(4)-S(3))/779.+S(3)
      ICHAR=D(1)
      TYPE 1,31
      RETURN
      END
```

```
C HIST1.FOR
C UPDATE 15-DEC-78
C       PLOTTING HISTOGRAM OF VARIABLES FROM AN UNFORMATTED SEQUENTIAL
C       FILE.  ONE FLOATING POINT VARIABLE IS STORED PER RECORD.
        INTEGER K(500)
        REAL S(4)
        TYPE 1
1       FORMAT(' INPUT FILE*',$)
        CALL ASSIGN(1,,-1,'RDO')
4       TYPE 2
2       FORMAT(' VARIABLE RANGE, INTERVAL')
        ACCEPT 3,XMIN,XMAX,XINT
        N=(XMAX-XMIN)/XINT
3       FORMAT(7F10.0)
        IF(N.LT.500)GO TO 5
        TYPE 6
6       FORMAT(' TOO MANY BINS')
        GO TO 4
5       TYPE 13
13      FORMAT(' MAX*',$)
        ACCEPT 14,MAX
14      FORMAT(I6)
        REWIND 1
        DO 9 I=1,N
9       K(I)=0
        EC=0.
7       READ(1,END=8)X
        IF(X.LT.XMIN.OR.X.GT.XMAX)GO TO 7
        EC=EC+1
        I=1+(X-XMIN)/XINT
        K(I)=K(I)+1
        GO TO 7
8       DO 10 I=1,N
10      IF(K(I).GT.MAX)MAX=K(I)
        S(1)=XMIN
        S(2)=XMAX
        S(3)=0.
        S(4)=MAX
        CALL SCALE(S)
        XL=XMIN
        YL=0.
        DO 11 I=1,N
        X=XMIN+I*XINT
        Y=K(I)
        CALL VECTOR(S,XL,YL,XL,Y)
        CALL VECTOR(S,XL,Y,X,Y)
        XL=X
11      YL=Y
        CALL VECTOR(S,XL,YL,XL,0.)
        CALL LET(S,XMIN,S(4))
        TYPE 15,EC
15      FORMAT('  NUMBER OF EVENTS:', F10.0)
        ACCEPT 12,I
12      FORMAT(72A1)
        GO TO 4
        END
```

```
C       IHIST.FOR
C       PLOTTING AN INTEGRAL DISTRIBUTION OF DATA STORED IN
C       UNFORMATTED SEQUENTIAL FILE.
        DIMENSION S(4),D(7000)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 10
10      FORMAT(' VARIABLE RANGE')
        ACCEPT 11,X1,X2
11      FORMAT(2F10.0)
        N=0
3       READ(1,END=2)X
        IF(N.EQ.6999)GO TO 2
        N=N+1
        D(N)=X
        GO TO 3
2       DO 4 I=2,N
        X=D(I)
        DO 7 J=I-1,1,-1
        IF(D(J).LT.X)GO TO 6
7       D(J+1)=D(J)
        D(1)=X
        GO TO 4
6       D(J+1)=X
4       CONTINUE
        S(1)=AMAX1(X1,D(1))
        S(2)=AMIN1(X2,D(N))
        S(3)=0.
        S(4)=N
        CALL SCALE(S)
        D(N+1)=D(N)
        DO 8 I=N,1,-1
        Y=N-I
        IF(D(I+1).GT.S(2).OR.D(I).LT.X1)GO TO 8
        CALL VECTOR(S,D(I),Y,D(I),Y+1.)
        CALL VECTOR(S,D(I),Y,D(I+1),Y)
8       CONTINUE
        ACCEPT 9,I
9       FORMAT(A2)
        END
```

```
      SUBROUTINE LET (S,A,B)
      DIMENSION S(4)
      IF (A.LT.S(1).OR.A.GT.S(2).OR.B.LT.S(3).OR.B.GT.S(4))
    1 RETURN
      X=1023.*(A-S(1))/(S(2)-S(1))-5.
      Y=779.*(B-S(3))/(S(4)-S(3))-7.
      IX=INT(X/32.)
      IY=INT(Y/32.)
      TYPE 1,29,32+IY,96+INT(Y)-32*IY,32+IX,64+INT(X)-32*IX,31,11
    1 FORMAT(1HS,7A1)
      RETURN
      END
```

```
C       LSF.FOR
C       LINEAR LEAST SQUARES FIT
        DIMENSION A(20),B(20),H(20,20),F(20),S(4)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
2       REWIND 1
        TYPE 3
3       FORMAT(' NUMBER OF PARAMATERS')
6       ACCEPT 4,N
        DO 7 I=1,N
        B(I)=0.
        DO 7 J=I,N
7       H(I,J)=0.
        SY=0.
        M=0
        S(1)=1.E37
        DO 15 I=2,4
15      S(I)=-S(I-1)
4       FORMAT(7I10)
9       READ(1,END=5)X,Y
        S(1)=AMIN1(S(1),X)
        S(2)=AMAX1(S(2),X)
        S(3)=AMIN1(S(3),Y)
        S(4)=AMAX1(S(4),Y)
        CALL FX(N,X,F)
        DO 8 I=1,N
        B(I)=B(I)+F(I)*Y
        DO 8 J=I,N
8       H(I,J)=H(I,J)+F(I)*F(J)
        M=M+1
        SY=SY+Y*Y
        GO TO 9
5       A(N)=B(N)
        DO 10 I=1,N-1
        A(I)=B(I)
        DO 10 J=I+1,N
10      H(J,I)=H(I,J)
        CALL MATINV(H,N,A,1,DET)
        DO 11 I=1,N
11      SY=SY-A(I)*B(I)
        SY=SY/(M-N)
        TYPE 12,SQRT(SY)
        TYPE 12,(A(I),I=1,N)
        DO 22 I=1,N
22      TYPE 12,(H(I,J)*SY,J=1,I)
12      FORMAT(1X,5G14.7)
13      REWIND 1
25      TYPE 24
24      FORMAT(' X= (999 FOR GRAPH)')
        ACCEPT 23,X
23      FORMAT(7F10.0)
17      FORMAT(A2)
        IF(X.EQ.999.)GO TO 28
        CALL FX(N,X,F)
        Y=0.
        YP=0.
        DO 26 J=1,N
26      Y=Y+A(J)*F(J)
        DO 29 J=N,2,-1
29      YP=YP*X+(J-1)*A(J)
        TYPE 27,Y,YP
27      FORMAT(' Y,DY/DX = ',2G14.7)
        GO TO 25
```

```
C       LSFCAL.FOR
C       LINEAR LEAST SQUARES FIT
C       WITH MODIFICATIONS TO BE USED FOR DETERMINING CALIBRATION
C       PARAMETERS.
        DIMENSION A(20),B(20),H(20,20),F(20),S(4)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
31      TYPE 30
30      FORMAT(' V0')
        ACCEPT 23,V0
2       REWIND 1
        TYPE 3
3       FORMAT(' NUMBER OF PARAMATERS')
6       ACCEPT 4,N
        IF(N.EQ.0)GO TO 31
        DO 7 I=1,N
        B(I)=0.
        DO 7 J=I,N
7       H(I,J)=0.
        SY=0.
        M=0
        S(1)=1.E37
        DO 15 I=2,4
15      S(I)=-S(I-1)
4       FORMAT(7I10)
9       READ(1,END=5)X,Y
        IF(Y.LE.V0)GO TO 9
        X=ALOG(X)
        Y=ALOG(Y-V0)
        S(1)=AMIN1(S(1),X)
        S(2)=AMAX1(S(2),X)
        S(3)=AMIN1(S(3),Y)
        S(4)=AMAX1(S(4),Y)
        CALL FX(N,X,F)
        DO 8 I=1,N
        B(I)=B(I)+F(I)*Y
        DO 8 J=I,N
8       H(I,J)=H(I,J)+F(I)*F(J)
        M=M+1
        SY=SY+Y*Y
        GO TO 9
5       A(N)=B(N)
        DO 10 I=1,N-1
        A(I)=B(I)
        DO 10 J=I+1,N
10      H(J,I)=H(I,J)
        CALL MATINV(H,N,A,1,DET)
        DO 11 I=1,N
11      SY=SY-A(I)*B(I)
        SY=SY/(M-N)
        TYPE 12,SQRT(SY)
        TYPE 12,(A(I),I=1,N)
        DO 22 I=1,N
22      TYPE 12,(H(I,J)*SY,J=1,I)
12      FORMAT(1X,5G14.7)
13      REWIND 1
25      TYPE 24
24      FORMAT(' X= (999 FOR GRAPH)')
        ACCEPT 23,X
23      FORMAT(7F10.0)
17      FORMAT(A2)
        IF(X.EQ.999.)GO TO 28
        CALL FX(N,X,F)
```

```
        Y=0.
        YP=0.
        DO 26 J=1,N
26      Y=Y+A(J)*F(J)
        DO 29 J=N,2,-1
29      YP=YP*X+(J-1)*A(J)
        TYPE 27,Y,YP
27      FORMAT(' Y,DY/DX = ',2G14.7)
        GO TO 25
28      CALL SCALE(S)
18      READ(1,END=19)X,Y
        IF(Y.LE.V0)GO TO 18
        X=ALOG(X)
        Y=ALOG(Y-V0)
        CALL LET(S,X,Y)
        TYPE 20
20      FORMAT(' 0')
        GO TO 18
19      X0=(S(1)+.15*S(2))/1.15
        DX=.01*(S(2)-X0)
        DO 14 I=0,100
        X=X0+I*DX
        Y=0.
        CALL FX(N,X,F)
        DO 21 J=1,N
21      Y=Y+A(J)*F(J)
        IF(I.GT.0)CALL VECTOR(S,XL,YL,X,Y)
        XL=X
14      YL=Y
        ACCEPT 17,I
        GO TO 2
        END
```

```
C       LSFCL.FOR
C       CLEANING UP DATA FOR LSF.
        DIMENSION S(4),X(1000),Y(1000)
        S(1)=1.E37
        DO 5 I=2,4
5       S(I)=-S(I-1)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'B',-1,'NEW')
        N=1
3       READ(1,END=4)X(N),Y(N)
        S(1)=AMIN1(S(1),X(N))
        S(2)=AMAX1(S(2),X(N))
        S(3)=AMIN1(S(3),Y(N))
        S(4)=AMAX1(S(4),Y(N))
        N=N+1
        GO TO 3
4       N=N-1
        CALL SCALE(S)
        DO 6 I=1,N
        CALL LET(S,X(I),Y(I))
6       TYPE 7
7       FORMAT(' O')
8       CALL HAIRS(S,U,V,IT)
        IF(IT.EQ.69)GO TO 9
        DM=1.E37
        DO 10 I=1,N
        D=ABS(U-X(I))+ABS(V-Y(I))
        IF(D.GE.DM)GO TO 10
        DM=D
        IM=I
10      CONTINUE
        CALL LET(S,X(IM),Y(IM))
        TYPE 11
11      FORMAT(' X')
        IF(IM.EQ.N)GO TO 13
        DO 12 I=IM+1,N
        X(I-1)=X(I)
12      Y(I-1)=Y(I)
13      N=N-1
        GO TO 8
9       DO 14 I=1,N
14      WRITE(2)X(I),Y(I)
        END
```

```
C       LSFCL.FOR
C       CLEANING UP DATA FOR LSF.
        DIMENSION S(4),X(1000),Y(1000)
        S(1)=1.E37
        DO 5 I=2,4
5       S(I)=-S(I-1)
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(2,'B',-1,'NEW')
        N=1
3       READ(1,END=4)X(N),Y(N)
        S(1)=AMIN1(S(1),X(N))
        S(2)=AMAX1(S(2),X(N))
        S(3)=AMIN1(S(3),Y(N))
        S(4)=AMAX1(S(4),Y(N))
        N=N+1
        GO TO 3
4       N=N-1
        CALL SCALE(S)
        DO 6 I=1,N
        CALL LET(S,X(I),Y(I))
6       TYPE 7
7       FORMAT(' O')
8       CALL HAIRS(S,U,V,IT)
        IF(IT.EQ.69)GO TO 9
        DM=1.E37
        DO 10 I=1,N
        D=ABS(U-X(I))+ABS(V-Y(I))
        IF(D.GE.DM)GO TO 10
        DM=D
        IM=I
10      CONTINUE
        CALL LET(S,X(IM),Y(IM))
        TYPE 11
11      FORMAT(' X')
        IF(IM.EQ.N)GO TO 13
        DO 12 I=IM+1,N
        X(I-1)=X(I)
12      Y(I-1)=Y(I)
13      N=N-1
        GO TO 8
9       DO 14 I=1,N
14      WRITE(2)X(I),Y(I)
        END
```

```
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2))630,710,630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
```

```
C       POPL.FOR
C       PLOTTING X,Y POINTS CONTAINED IN SEQUENTIAL UNFORMATTED FILE.
C       ONE X,Y POINT PER RECORD (FLOATING PT. X AND Y).
        DIMENSION S(4)
11      TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        S(1)=1.E37
        DO 2 I=2,4
2       S(I)=-S(I-1)
3       READ(1,END=4)X,Y
        S(1)=AMIN1(S(1),X)
        S(2)=AMAX1(S(2),X)
        S(3)=AMIN1(S(3),Y)
        S(4)=AMAX1(S(4),Y)
        GO TO 3
4       TYPE 700,(S(I),I=1,4)
41      TYPE 701
        ACCEPT 500,I
        IF(I.EQ.0) GOTO 42
        TYPE 702
        ACCEPT 501,S(I)
        GOTO 41
42      REWIND 1
        CALL SCALE(S)
        EC=0.
5       READ(1,END=6)X,Y
        EC=EC+1
        CALL BPOINT(S,X,Y)
        GO TO 5
6       CALL LET(S,.85*S(1)+.15*S(2),.05*S(3)+.95*S(4))
        TYPE 703,EC
        ACCEPT 500,IDUMMY
        CALL CLOSE(1)
        GOTO 11

500     FORMAT(I6)
501     FORMAT(F10.0)
700     FORMAT(' XMIN:',E12.5,' XMAX:',E12.5/
    *' YMIN:',E12.5,' YMAX:',E12.5/)
701     FORMAT('+MODIFY 1,2,3,4,0=EXIT *',$)
702     FORMAT('+*',$)
703     FORMAT('+ NUMBER OF EVENTS:',F10.0)
        END
```

```
C       POPL.FOR
C       PLOTTING X,Y POINTS CONTAINED IN SEQUENTIAL UNFORMATTED FILE.
C       ONE X,Y POINT PER RECORD (FLOATING PT. X AND Y).
        DIMENSION S(4)
11      TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1)
        S(1)=1.E37
        DO 2 I=2,4
2       S(I)=-S(I-1)
3       READ(1,END=4)X,Y
        S(1)=AMIN1(S(1),X)
        S(2)=AMAX1(S(2),X)
        S(3)=AMIN1(S(3),Y)
        S(4)=AMAX1(S(4),Y)
        GO TO 3
4       TYPE 700,(S(I),I=1,4)
41      TYPE 701
        ACCEPT 500,I
        IF(I.EQ.0) GOTO 42
        TYPE 702
        ACCEPT 501,S(I)
        GOTO 41
42      REWIND 1
        CALL SCALE(S)
        EC=0.
5       READ(1,END=6)X,Y
        EC=EC+1
        CALL BPOINT(S,X,Y)
        GO TO 5
6       CALL LET(S,.85*S(1)+.15*S(2),.05*S(3)+.95*S(4))
        TYPE 703,EC
        ACCEPT 500,IDUMMY
        CALL CLOSE(1)
        GOTO 11

500     FORMAT(I6)
501     FORMAT(F10.0)
700     FORMAT(' XMIN:',E12.5,' XMAX:',E12.5/
    *' YMIN:',E12.5,' YMAX:',E12.5/)
701     FORMAT('+MODIFY 1,2,3,4,0=EXIT *',$)
702     FORMAT('+*',$)
703     FORMAT('+ NUMBER OF EVENTS:',F10.0)
        END
```

```
C       SURFA.FOR
C       SUBROUTINE TO COMPUTE THE AREA AND APPROXIMATE PERIMETER
C       OF THE SURFACE OPENING OF TRACKS, AREA AND PERIM,
C       AS A FUNCTION OF THE FRACTION OF THE LAYER REMOVED, RHO,
C       THE DIP ANGLE, DELTA, AND THE CONE ANGLE, THETA.  THE COMPUTED
C       AREA AND PERIMENTER ARE IN THE UNITS OF LAYER THICKNESS
C       SQUARED AND LAYER THICKNESS.
        SUBROUTINE ETA(RHO,DELTA,THETA,AREA,PERIM)
        BULK=RHO
        SD=SIN(DELTA)
        ST=SIN(THETA)
        A=BULK*COS(THETA)/(SD+ST)
        B=BULK*SQRT((SD-ST)/(SD+ST))
        PI=3.1415927
        AREA=PI*A*B
        PERIM=2.*PI*SQRT(.5*(A*A+B*B))
        RETURN
        END
```

```
C       TRAN.FOR
C       SUBROUTINE TO CAUSE TRANSLATION OF DX, DY MM.
        SUBROUTINE TRAN(IA,DX,DY)
        DIMENSION IA(45)
        LX=.5+100.*ABS(DX)
        LY=.5+100.*ABS(DY)
        IF(DY.LT.0.)LY=LY+100
        IF(DX.LT.0.)LX=LX+100
        IX=0
        IY=0
        IF(DX)1,2,3
1       IX="400
        GO TO 2
3       IX="1000
2       IF(DY)4,5,6
4       IY="2000
        GO TO 5
6       IY="4000
5       IB=IX+IY
        CALL TR(IA,MIN0(LX,LY),IB)
        IF(LX-LY)7,8,9
7       CALL TR(IA,LY-LX,IY)
8       IF(DY.LT.0..AND.DX.GE.0.)CALL TR(IA,100,"4000)
        IF(DX.LT.0..AND.DY.GE.0.)CALL TR(IA,100,"1000)
        IF(DX.LT.0..AND.DY.LT.0.)CALL TR(IA,100,"5000)
        RETURN
9       CALL TR(IA,LX-LY,IX)
        GO TO 8
        END
C
C       INITIALIZING FOR THE STAGE TRANSLATION.
C
        SUBROUTINE TRANI(IA)
        DIMENSION IA(45)
        DO 1 I=1,45
1       IA(I)=.5+20.*SQRT(45./FLOAT(I))
        RETURN
        END
C
C       SENDING CHARACTERS.
C
        SUBROUTINE TR(IA,N,K)
        DIMENSION IA(45)
        IF(N.EQ.0)RETURN
        DO 1 I=1,N
        IP=MIN0(45,I,N-I+1)
        CALL IPOKE("167772,K)
        DO 2 J=1,20
2       CONTINUE
        CALL IPOKE("167772,0)
        DO 1 J=1,IA(IP)
1       CONTINUE
        RETURN
        END
```

```
C       TRAN.FOR
C       SUBROUTINE TO CAUSE TRANSLATION OF DX, DY MM.
        SUBROUTINE TRAN(IA,DX,DY)
        DIMENSION IA(45)
        LX=.5+100.*ABS(DX)
        LY=.5+100.*ABS(DY)
        IF(DY.LT.0.)LY=LY+100
        IF(DX.LT.0.)LX=LX+100
        IX=0
        IY=0
        IF(DX)1,2,3
1       IX="400
        GO TO 2
3       IX="1000
2       IF(DY)4,5,6
4       IY="2000
        GO TO 5
6       IY="4000
5       IB=IX+IY
        CALL TR(IA,MIN0(LX,LY),IB)
        IF(LX-LY)7,8,9
7       CALL TR(IA,LY-LX,IY)
8       IF(DY.LT.0..AND.DX.GE.0.)CALL TR(IA,100,"4000)
        IF(DX.LT.0..AND.DY.GE.0.)CALL TR(IA,100,"1000)
        IF(DX.LT.0..AND.DY.LT.0.)CALL TR(IA,100,"5000)
        RETURN
9       CALL TR(IA,LX-LY,IX)
        GO TO 8
        END
C
C       INITIALIZING FOR THE STAGE TRANSLATION.
C
        SUBROUTINE TRANI(IA)
        DIMENSION IA(45)
        DO 1 I=1,45
1       IA(I)=.5+20.*SQRT(45./FLOAT(I))
        RETURN
        END
C
C       SENDING CHARACTERS.
C
        SUBROUTINE TR(IA,N,K)
        DIMENSION IA(45)
        IF(N.EQ.0)RETURN
        DO 1 I=1,N
        IP=MIN0(45,I,N-I+1)
        CALL IPOKE("167772,K)
        DO 2 J=1,20
2       CONTINUE
        CALL IPOKE("167772,0)
        DO 1 J=1,IA(IP)
1       CONTINUE
        RETURN
        END
```

```
      SUBROUTINE VECTOR(S,A,B,C,D)
      DIMENSION S(4)
      IF (A.LT.S(1).OR.C.LT.S(1).OR.A.GT.S(2).OR.
     1C.GT.S(2).OR.B.LT.S(3).OR.D.LT.S(3).
     2OR.B.GT.S(4).OR.D.GT.S(4))RETURN
      X1=1023.*(A-S(1))/(S(2)-S(1))
      X2=1023.*(C-S(1))/(S(2)-S(1))
      Y1=779.*(B-S(3))/(S(4)-S(3))
      Y2=779.*(D-S(3))/(S(4)-S(3))
      LY1=Y1
      LY2=Y2
      IX1=INT(X1/32.)
      IX2=INT(X2/32.)
      IY1=INT(Y1/32.)
      IY2=INT(Y2/32.)
      LX1=64+INT(X1)-32*IX1
      TYPE 1,29,32+IY1,96+LY1-32*IY1,32+IX1,LX1
     *,32+IY2,96+LY2-32*IY2,32+IX2,64+INT(X2)-32*IX2
1     FORMAT('+',72A1)
      RETURN
      END
```

```
C       XYSORT.FOR
C       MERGE SORT OF FILES CONTAINING TWO FLOATING POINT VARIABLES
C       PER RECORD.  THE FIRST VARIABLE IS THE KEY.
C
        TYPE 1
1       FORMAT(' INPUT FILE NAME'/)
        CALL ASSIGN(8,'A',-1,'RDO')
        TYPE 2
2       FORMAT(' OUTPUT FILE NAME'/)
        CALL ASSIGN(1,'A',-1,'NEW')
        DO 3 I=2,4
        TYPE 4,I
4       FORMAT(' TEMPORARY FILE #',I1,' NAME'/)
3       CALL ASSIGN(I,'A',-1,'SCR')
        N=0
        J=2
6       READ(8,END=5)XS,YS
        N=N+1
        J=3-J
        WRITE(J)XS,YS
        GO TO 6
5       ENDFILE 1
        ENDFILE 2
        CALL RW
        IP=1.D0+DLOG(DBLE(FLOAT(N)-.5))/DLOG(2.D0)
        K=-1
        IF(MOD(IP,2).EQ.0)GO TO 10
        K=1
        DO 11 I=1,2
13      READ(I,END=11)XS,YS
        II=I+2
        WRITE(II)XS,YS
        GO TO 13
11      CONTINUE
        ENDFILE 3
        ENDFILE 4
        CALL RW
10      J=1
        TYPE 12,N,IP
12      FORMAT(1X,7I10)
        DO 7 I=1,IP
        TYPE 12,I,J
        IF(K.EQ.1)GO TO 8
        CALL MTAPE(1,2,3,4,J,N)
        GO TO 9
8       CALL MTAPE(3,4,1,2,J,N)
9       K=-K
7       J=2*J
        END
C
        SUBROUTINE RW
        DO 1 I=1,4
1       REWIND I
        RETURN
        END
C
        SUBROUTINE MTAPE(IA,IB,IC,ID,M,N)
        L=1+(N-1)/(2*M)
        J=1
        DO 1 I=1,L
        IF(J.EQ.1)GO TO 2
        CALL MBLOCK(IA,IB,ID,M)
        GO TO 1
2       CALL MBLOCK(IA,IB,IC,M)
```

```
1       J=-J
        ENDFILE IC
        ENDFILE ID
        CALL RW
        RETURN
        END
C
        SUBROUTINE MBLOCK(IA,IB,IC,M)
        JA=1
        JB=1
        IFLAG=0
        READ(IA,END=6)XS,YS
8       READ(IB,END=7)XT,YT
        IF(IFLAG.EQ.1)GO TO 4
1       IF(XS.LE.XT)GO TO 2
        WRITE(IC)XT,YT
        IF(JB.EQ.M)GO TO 3
        JB=JB+1
        READ(IB,END=3)XT,YT
        GO TO 1
2       WRITE(IC)XS,YS
        IF(JA.EQ.M)GO TO 4
        JA=JA+1
        READ(IA,END=4)XS,YS
        GO TO 1
3       WRITE(IC)XS,YS
        IF(JA.EQ.M)GO TO 5
        JA=JA+1
        READ(IA,END=5)XS,YS
        GO TO 3
4       WRITE(IC)XT,YT
        IF(JB.EQ.M)GO TO 5
        JB=JB+1
        READ(IB,END=5)XT,YT
        GO TO 4
5       RETURN
6       IFLAG=1
        GO TO 8
7       CONTINUE
        IF(IFLAG)5,3,5
        END
```

REFERENCES

1. Benton E. V. and Henke R. P. (1983) Radiation exposures during spaceflight and their measurement. Adv. Space Res. 3, 171.

2. Janni J. (1969) A review of Soviet manned space flight dosimetry results. Aerospace Medicine 40, 1547.

3. Petrov V., Akatov Y., Kozlova S., Markelov V., Nesterov V., Redko V., Smirenny L., Khortsev A. and Chernikh I. (1975) The study of the radiation environment in near-Earth space. Space Res. 13, 129.

4. Bailey J. V. (1977) Dosimetry during space missions. IEEE Trans. Nucl. Sci., NS-23, #4, 1379.

5. Jordan, T. M. (1983) Radiation protection for manned space activities, JPL Publication 83-26, Jet Propulsion Laboratory, Pasadena.

6. Stassinopoulos, E. G. (1980) The geostationary radiation environment. J. Spacecraft and Rockets 17, 145.

7. Watts J. W. Jr. and Wright J. J. (1976) Charged particle radiation environment for the Spacelab and other missions in low Earth orbit--Revision A. NASA Tech. Memo. TMX-73358, 1-137.

8. Plato P. and Hudson G. (1980) Performance testing of personnel dosimetry service procedures manual. NUREG CR-1063, January 1980.

9. Roberts J. H., Parker R. A., Congel F. J., Kastner J. and Oltman B. G. (1970) Environmental neutron measurements with solid state track recorders. Rad. Effects 3, 283-285.

10. Benton E. V., Cassou R. M., Frank A. L. and Henke R. P. (1979) Space radiation dosimetry on board Cosmos 936: U.S. portion of Experiment K-206. Final Report, NASA Contract No. NAS2-9504.

11. Benton E. V., Henke R. P., Frank A. L., Johnson C., Cassou R., Tran M., and Etter E. (1980) Space radiation aboard Cosmos 1129: U. S. portion of Experiment K-309. Final Science Report, NASA Contract No. NAS2-10151.

12. Heinrich W. (1982) Private communication, University of Siegen, Siegen, Germany.